Scheduling in IaaS Cloud Computing Environments: Anything New?

Alexandru Iosup
Parallel and Distributed Systems Group
TU Delft

Hebrew University of Jerusalem, Israel
June 6, 2013

Lectures in Israel, mostly at the Technion Computer Engineering (TCE) Center

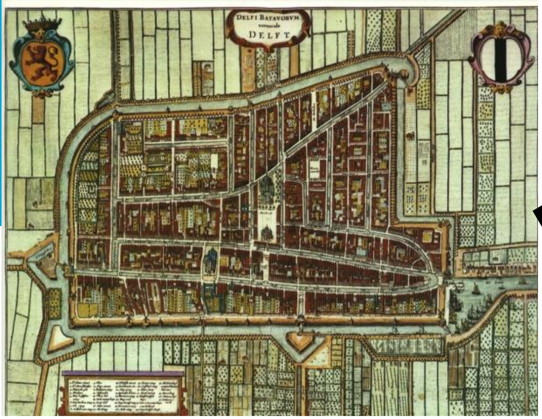
IaaS Cloud Benchmarking	May 7
Massivizing Online Social Games	May 9
Gamification in Higher Education	May 27
Lectures at IBM Haifa, Intel Haifa	June 2,3
Scheduling in IaaS Clouds	**HUJI June 5**

A TU Delft perspective on Big Data Processing and Preservation **June 6** **10am Taub 337**

Grateful to Orna Agmon Ben-Yehuda, Assaf Schuster, Isaac Keslassy.
Thanks to Dror Feitelson. Also thankful to Bella Rotman and Ruth Boneh.

Thanks to Michael Factor, Ronny Ronen.

(TU) Delft – the Netherlands – Europe



founded 13th century
pop: 100,000

founded 1842
pop: 13,000

pop: 16.5 M

(We are here) אנחנו כאן

The Parallel and Distributed Systems Group at TU Delft

VENI

Alexandru Iosup

Grids/Clouds
P2P systems
Big Data
Online gaming

Dick Epema

Grids/Clouds
P2P systems
Video-on-demand
e-Science

VENI

Ana Lucia Varbanescu

HPC systems
Multi-cores
Big Data
e-Science

Henk Sips

HPC systems
Multi-cores
P2P systems

VENI

Johan Pouwelse

P2P systems
File-sharing
Video-on-demand

Home page

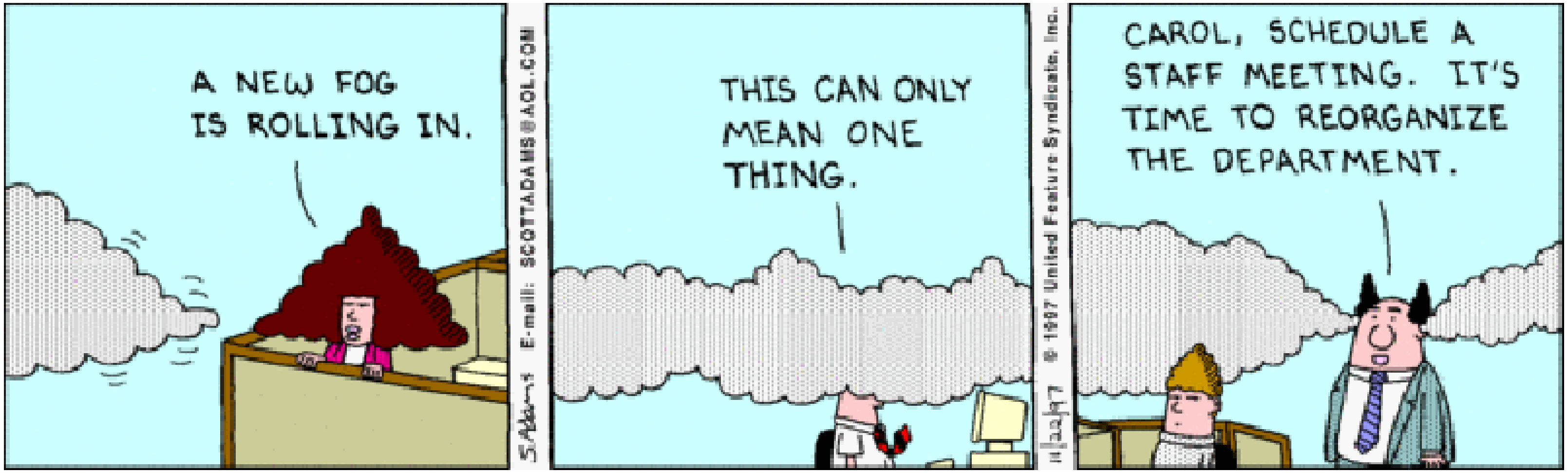
- www.pds.ewi.tudelft.nl

Publications

- see PDS publication database at publications.st.ewi.tudelft.nl

What is Cloud Computing?
1. A Cloudy Buzzword

- 18 definitions in computer science (ECIS'10). NIST has one. Cal has one. We have one.
- "We have redefined cloud computing to include everything that we already do." Larry Ellison, Oracle, 2009

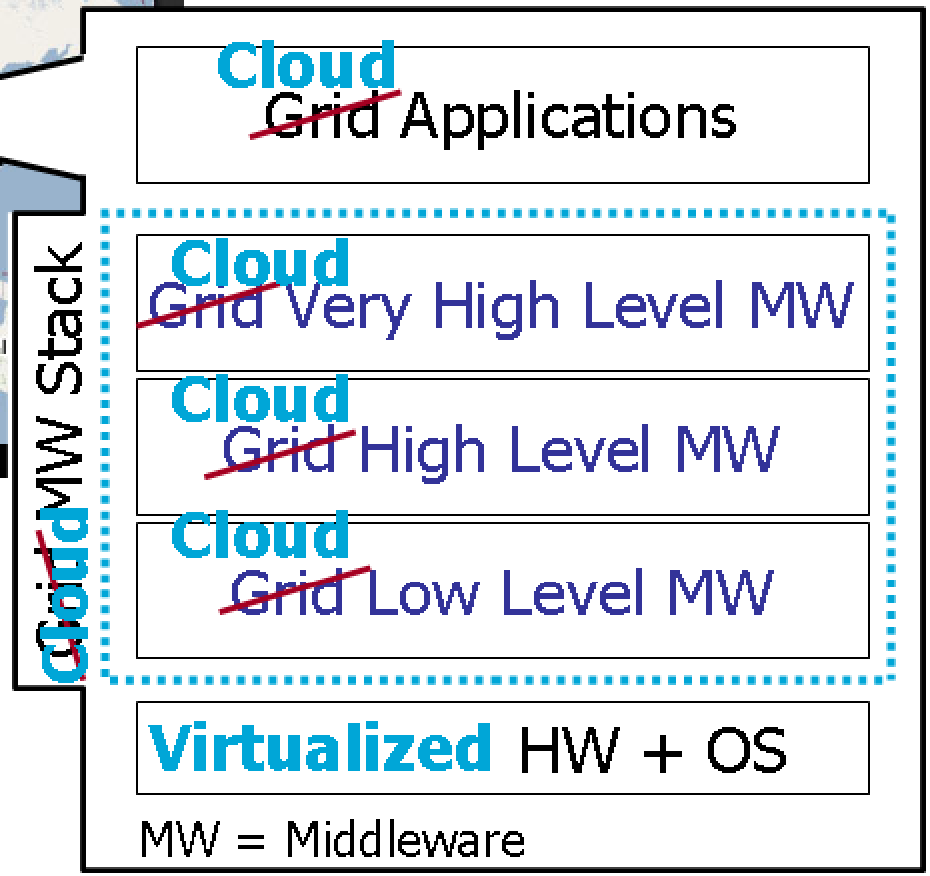


Source: http://dilbert.com/strips/comic/1997-11-22/

What is Cloud Computing?
2. A Descendant* of the Grid Idea

* Subset.

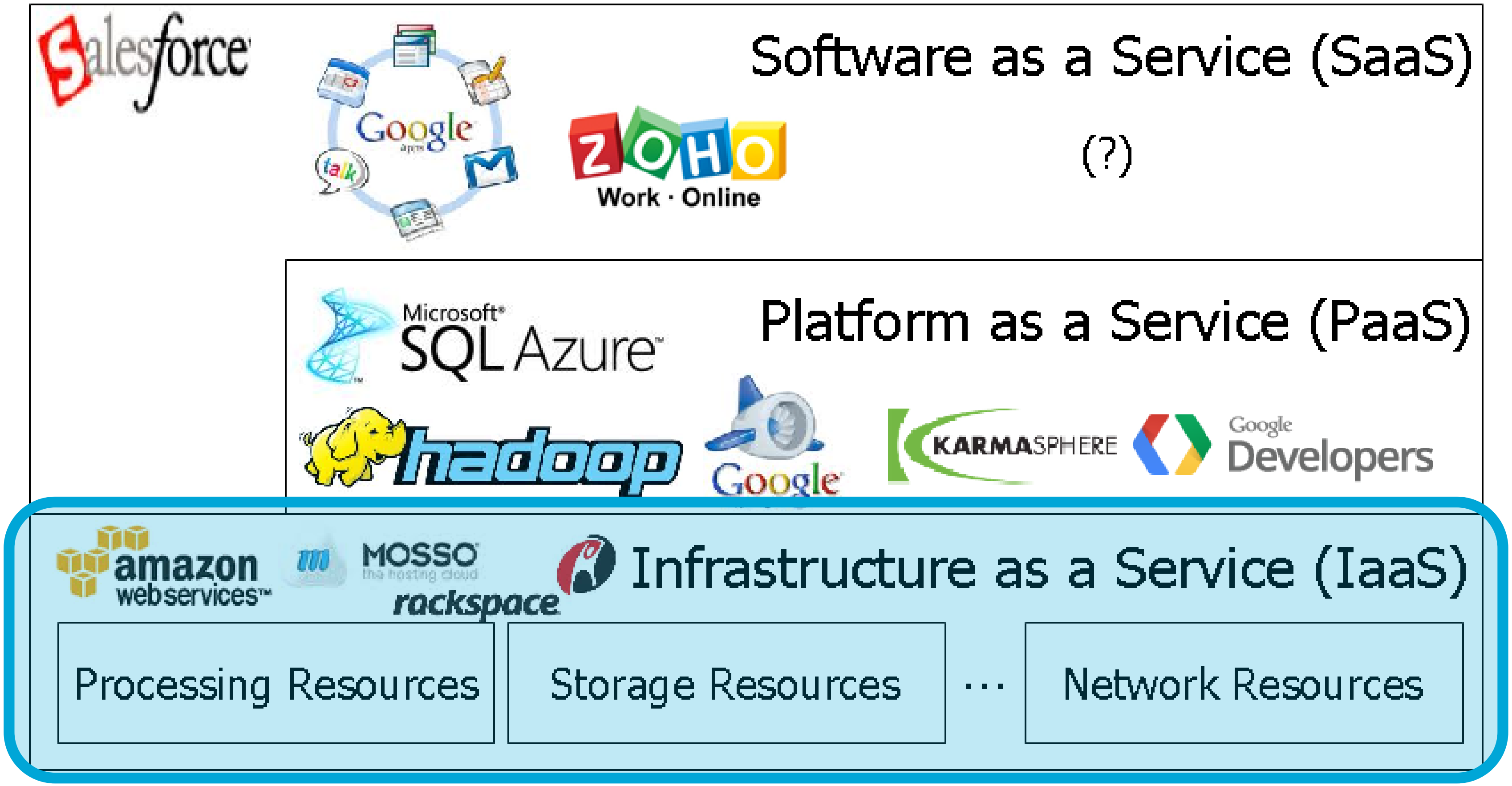
Source: http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/



"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [+ for] nontrivial QoS." I. Foster, 1998 + 1999

What is Cloud Computing?
3. A Useful IT Service

"Use only when you want! Pay only for what you use!"



Scheduling in IaaS Clouds
An Overview

Cloud operator:

Which resources to lease?
Where to place? Penalty v reward?

Uptime Institute
GREEN ENTERPRISE
IT AWARD 2010

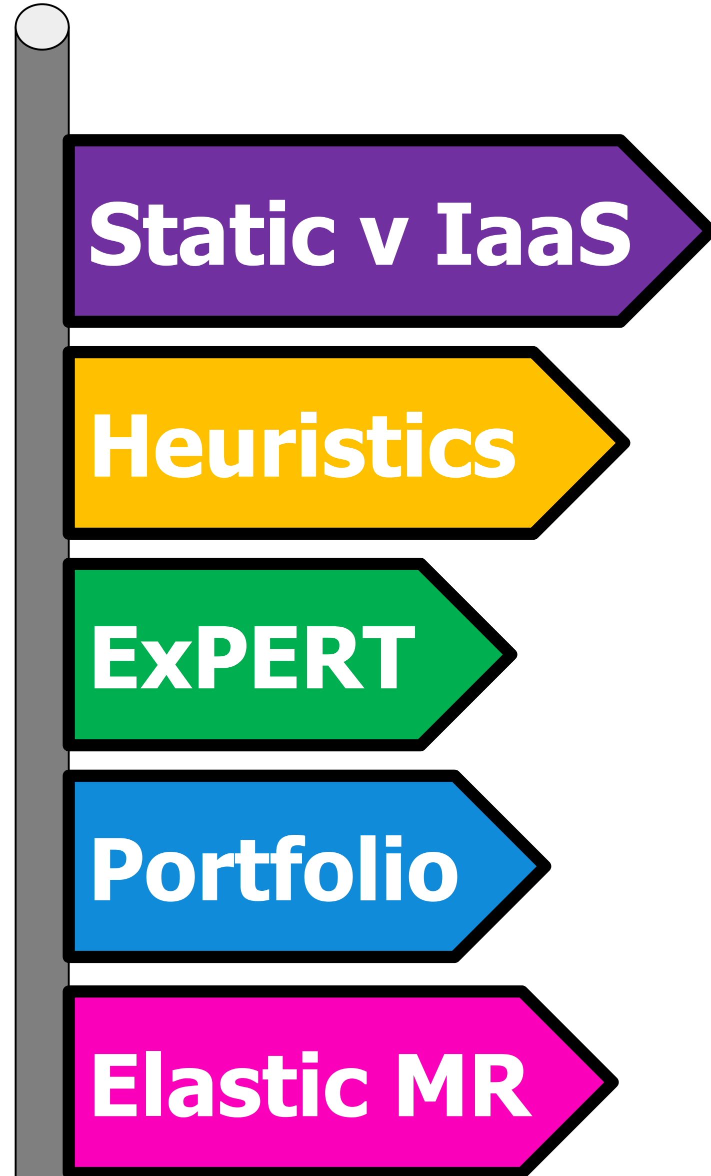
[redacted]

Cloud customer:

Which resources to lease?
When? How many? When stop?
Utility functions?

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
    1. **Static vs IaaS**
    2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
    3. **ExPERT Pareto-Optimal User-Sched.**
    4. **Portfolio Scheduling for Data Centers**
    5. **Elastic MapReduce**
3. Take-Home Message



Warm-Up Question: (2 minutes think-time + 2 minutes open discussion)

- Think about own experience
- Convince your partner before proposing an answer
- Tell everyone the answer

Q: How well would **your** workloads perform if executed on today's IaaS clouds?

What I'll Talk About

Real-World IaaS Cloud Performance and Implications on Many-Task Scientific Workloads

1. **Previous work**
2. **Experimental setup**
3. **Experimental results**
4. **Implications on Many-Task Scientific workloads**

Q: How well would previous many-task workloads perform if executed on today's IaaS clouds?

Some Previous Work
(>50 important references across our studies)

Virtualization Overhead

- Loss below 5% for computation [Barham03] [Clark04]
- Loss below 15% for networking [Barham03] [Menon05]
- Loss below 30% for parallel I/O [Vetter08]
- Negligible for compute-intensive HPC kernels [You06] [Panda06]

Cloud Performance Evaluation

- Performance and cost of executing a sci. workflows [Dee08]
- Study of Amazon S3 [Palankar08]
- Amazon EC2 for the NPB benchmark suite [Walker08] or selected HPC benchmarks [Hill08]
- CloudCmp [Li10]
- Kosmann et al.

Production IaaS Cloud Services

- **Production IaaS cloud:** lease resources (infrastructure) to users, operate on the market and have active customers

Name	Cores (ECUs)	RAM [GB]	Archi. [bit]	Disk [GB]	Cost [$/h]
Amazon EC2					
m1.small	1 (1)	1.7	32	160	0.1
m1.large	2 (4)	7.5	64	850	0.4
m1.xlarge	4 (8)	15.0	64	1,690	0.8
c1.medium	2 (5)	1.7	32	350	0.2
c1.xlarge	8 (20)	7.0	64	1,690	0.8
GoGrid (GG)					
GG.small	1	1.0	32	60	0.19
GG.large	1	1.0	64	60	0.19
GG.xlarge	3	4.0	64	240	0.76
Elastic Hosts (EH)					
EH.small	1	1.0	32	30	£0.042
EH.large	1	4.0	64	30	£0.09
Mosso					
Mosso.small	4	1.0	64	40	0.06
Mosso.large	4	4.0	64	160	0.24

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

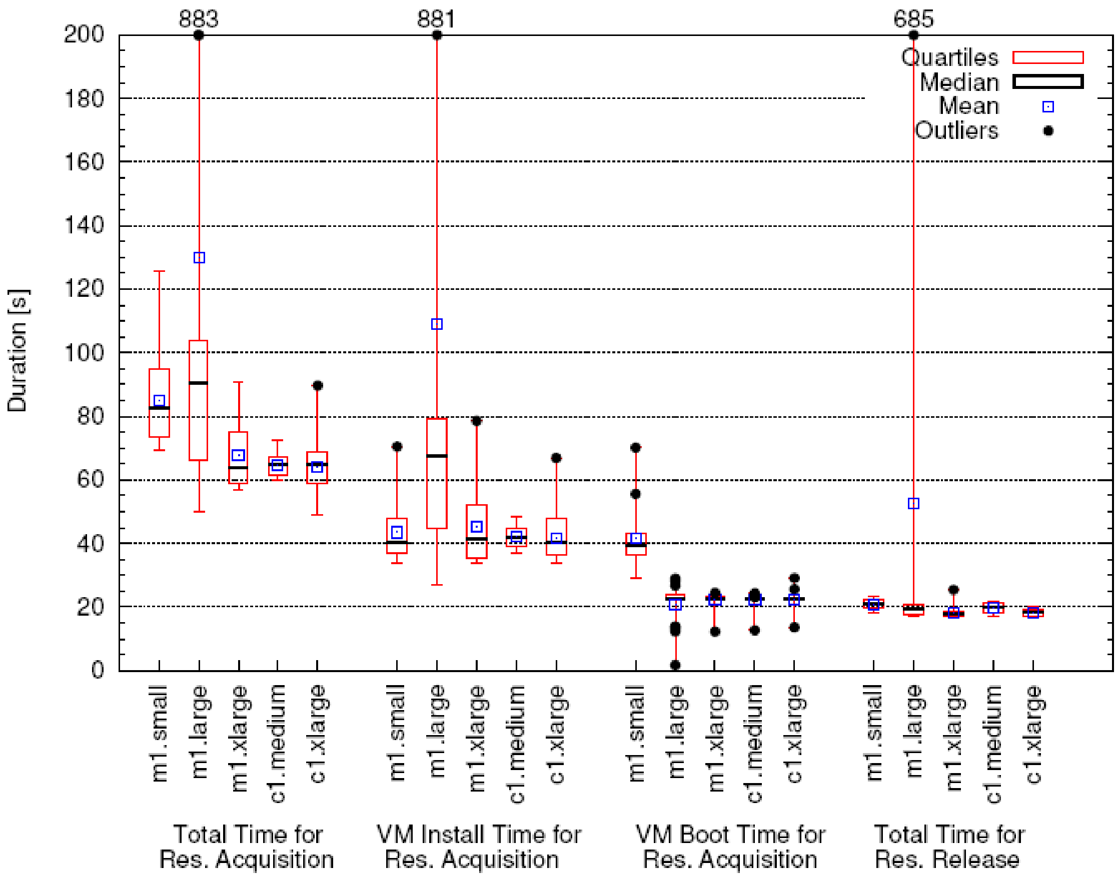
Our Method

- Based on general performance technique: model performance of individual components; system performance is performance of workload + model [Saavedra and Smith, ACM TOCS'96]
- Adapt to clouds:
  1. Cloud-specific elements: resource provisioning and allocation
  2. Benchmarks for single- and multi-machine jobs
  3. Benchmark CPU, memory, I/O, etc.:

Type	Suite/Benchmark	Resource	Unit
SI	LMbench/all [24]	Many	Many
SI	Bonnie/all [25], [26]	Disk	MBps
SI	CacheBench/all [27]	Memory	MBps
MI	HPCC/HPL [28], [29]	CPU	GFLOPS
MI	HPCC/DGEMM [30]	CPU	GFLOPS
MI	HPCC/STREAM [30]	Memory	GBps
MI	HPCC/RandomAccess [31]	Network	MUPS
MI	HPCC/b_{eff}(lat.,bw.) [32]	Comm.	μs, GBps

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

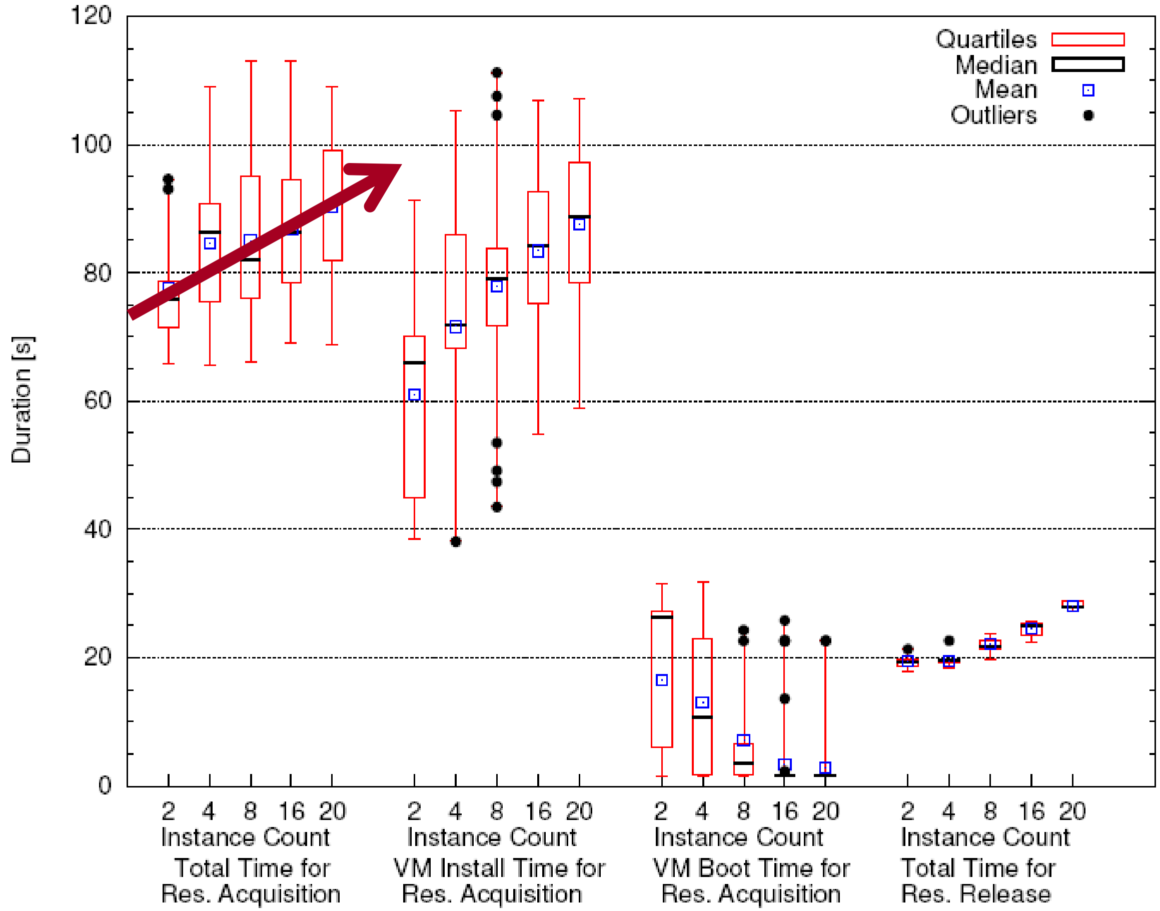
Leasing and Releasing Single Resource: Time Depends on Instancce Type



- Boot time non-negligible

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Multi-Resource: Time ~ O(log(#resources))

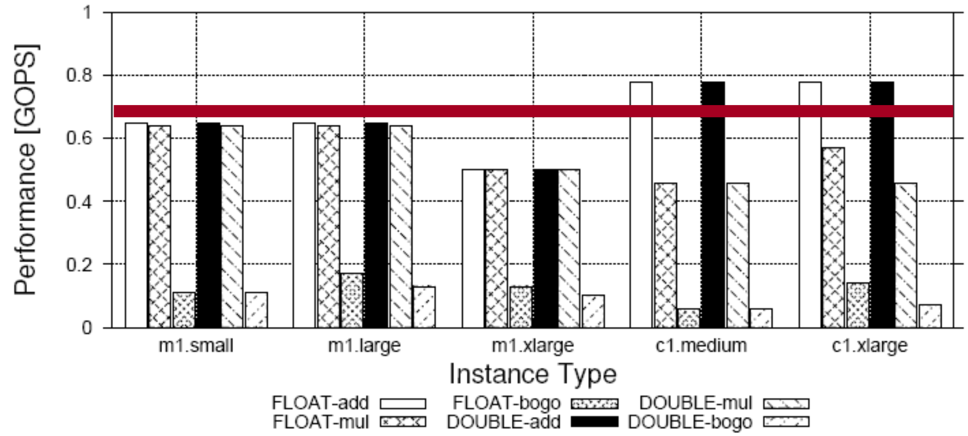


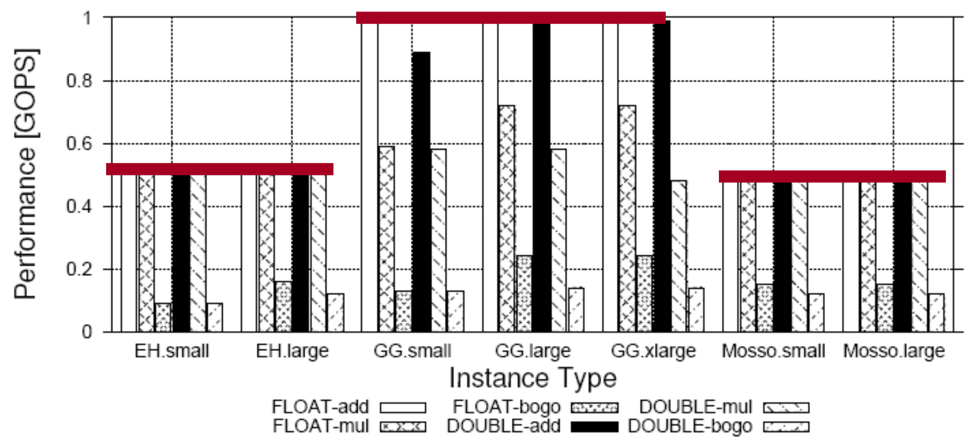
- Time for ***multi-***resource increases with number of resources

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

CPU Performance of Single Resource: ¼..1/7 Theoretical Peak

- ECU definition: "a 1.1 GHz 2007 Opteron" ~ 4 flops per cycle at full pipeline, which means at peak performance one ECU equals 4.4 gigaflops per second (GFLOPS)
- Real performance 0.6..0.1 GFLOPS = ~1/4..1/7 theoretical peak





Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Implications: Simulations

- Input: real-world workload traces, grids and PPEs
  - Selected BoTs

Trace ID, Source (Trace ID in Archive)	Trace Time [mo.]	Number of Jobs	Number of Users	System Size Sites	System Size CPUs	Load [%]
Grid Workloads Archive [13], 6 traces						
1. DAS-2 (1)	18	1.1M	333	5	0.4K	15+
2. RAL (6)	12	0.2M	208	1	0.8K	85+
3. GLOW (7)	3	0.2M	18	1	1.6K	60+
4. Grid3 (8)	18	1.3M	19	29	3.5K	-
5. SharcNet (10)	13	1.1M	412	10	6.8K	-
6. LCG (11)	1	0.2M	216	200+	24.4K	-
Parallel Workloads Archive [16], 4 traces						
7. CTC SP2 (6)	11	0.1M	679	1	0.4K	66
8. SDSC SP2 (9)	24	0.1M	437	1	0.1K	83
9. LANLO2K (10)	5	0.1M	337	1	2.0K	64
10. SDSC DS (19)	13	0.1M	460	1	1.7K	63

- Running in
  - Original env.
  - Cloud with source-like perf.
  - Cloud with measured perf. (model: 1/7)
- Metrics
  - WT, ReT, BSD(10s)
  - Cost [CPU-h]

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Implications: Clouds, Real Good for Immediate Work, Long-Run Costly

Trace ID	Source env. (Grid/PPI) AWT [s]	Source env. (Grid/PPI) AReT [s]	Source env. (Grid/PPI) ABSD (10s)	Cloud (real performance) AReT [s]	Cloud (real performance) ABSD (10s)	Cloud (real performance) Total Cost [CPU-h,M]	Cloud (source performance) AReT [s]	Cloud (source performance) ABSD (10s)	Cloud (source performance) Total Cost [CPU-h,M]
DAS-2	432	802	11	2,292	2.39	2	450	2	1.19
RAL	13,214	27,807	68	131,300	1	40	18,837	1	6.39
GLOW	9,162	17,643	55	59,448	1	3	8,561	1	0.60
Grid3	-	7,199	-	50,470	3	19	7,279	3	3.60
SharcNet	31,017	61,682	242	219,212	1	73	31,711	1	11.34
LCG	-	9,011	-	63,158	1	3	9,091	1	0.62
CTC SP2	25,748	37,019	78	75,706	1	2	11,351	1	0.30
SDSC SP2	26,705	33,388	389	46,818	2	1	6,763	2	0.16
LANL O2K	4,658	9,594	61	37,786	2	1	5,016	2	0.26
SDSC DS	32,271	33,807	516	57,065	2	2	6,790	2	0.25

- Cost:
  - Clouds, real >> Clouds, source
- Performance:
  - AReT: Clouds, real >> Clouds, source (**bad**)
  - AWT,ABSD: Clouds, real << Source env. (**good**)

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
   1. **Static vs IaaS**
   2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
   3. **ExPERT Pareto-Optimal User-Sched.**
   4. **Portfolio Scheduling for Data Centers**
   5. **Elastic MapReduce**
3. Take-Home Message

Warm-Up Question:
(2 minutes think-time + 2 minutes open discussion)

- Think about own experience
- Convince your partner before proposing an answer
- Tell everyone the answer

Q: How would **you** setup the provisioning and allocation policies for a particular IaaS cloud?

What I'll Talk About

Provisioning and Allocation Policies for Customers of IaaS Clouds

1. Online decisions via heuristics: an empirical study
    1. Experimental setup
    2. Experimental results
2. ExPERT : semi-offline computation + online assistance of cloud users

Heuristics

ExPERT

Provisioning and Allocation Policies*

* For User-Level Scheduling

- Provisioning

Policy	Class	Trigger	Adaptive
Startup	Static	–	–
OnDemand	Dynamic	QueueSize	No
ExecTime	Dynamic	Exec.Time	Yes
ExecAvg	Dynamic	Exec.Time	Yes
ExecKN	Dynamic	Exec.Time	Yes
QueueWait	Dynamic	Wait Time	Yes

- Allocation

Policy	Queue-based	Known job durations
FCFS	Yes	No
FCFS-NW	No	No
SJF	Yes	Yes

- Also looked at combined Provisioning + Allocation policies

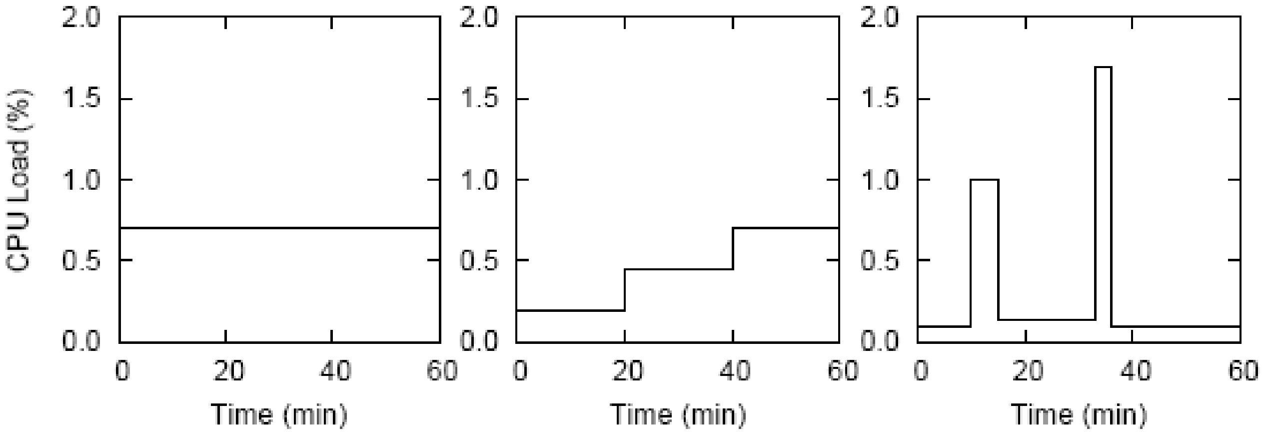
The SkyMark Tool for IaaS Cloud Benchmarking

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Experimental Setup (1)

- Environments
  - DAS4, Florida International University (FIU)
  - Amazon EC2
- Workloads
  - Bottleneck
  - Arrival pattern

Workload Unit	CPU	Memory	I/O	Appears in
WU1	X			WL1
WU2		X		WL2,WL4
WU3			X	WL3,WL4



Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid2012 + PDS Tech.Rep.2011-009

Experimental Setup (2)

- **Performance Metrics**
  - Traditional: Makespan, Job Slowdown
  - Workload Speedup One (SU1)
  - Workload Slowdown Infinite (SUinf)

$$SU_1(W) = \frac{MS(W)}{\sum_{i \in W} t_R(i)}$$

$$SU_\infty(W) = \frac{MS(W)}{\max_{i \in W} t_R(i)}$$

- **Cost Metrics**
  - Actual Cost (Ca)
  - Charged Cost (Cc)

$$C_a(W) = \sum_{i \in leased\ VMs} t_{stop}(i) - t_{start}(i)$$

$$C_c(W) = \sum_{i \in leased\ VMs} \lceil t_{stop}(i) - t_{start}(i) \rceil$$

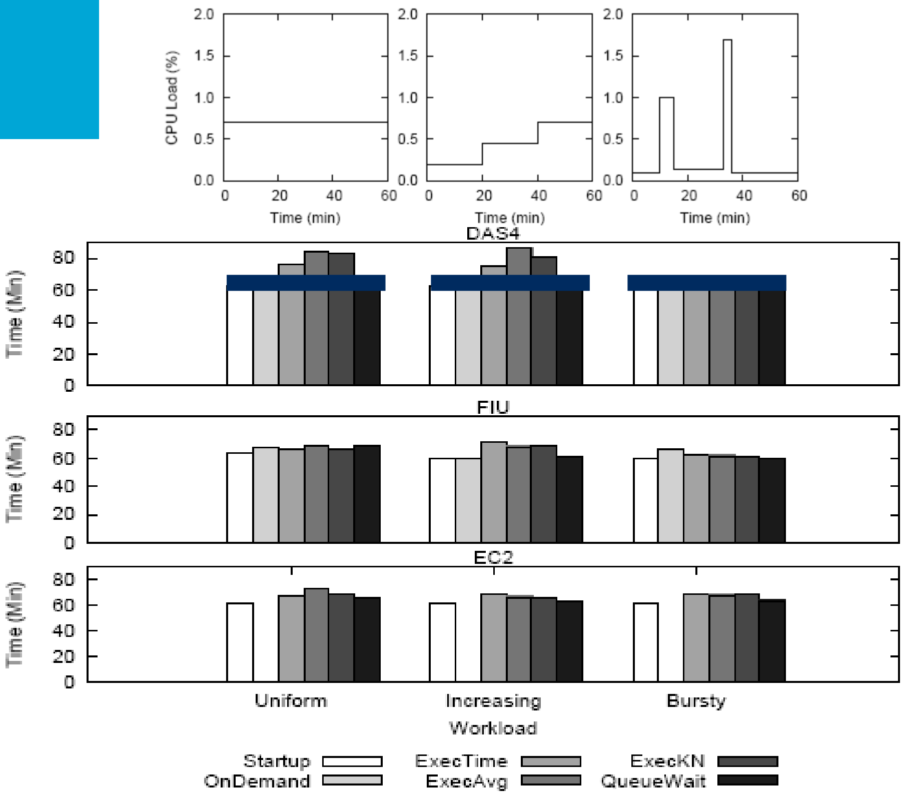
- **Compound Metrics**
  - Cost Efficiency (Ceff)
  - Utility

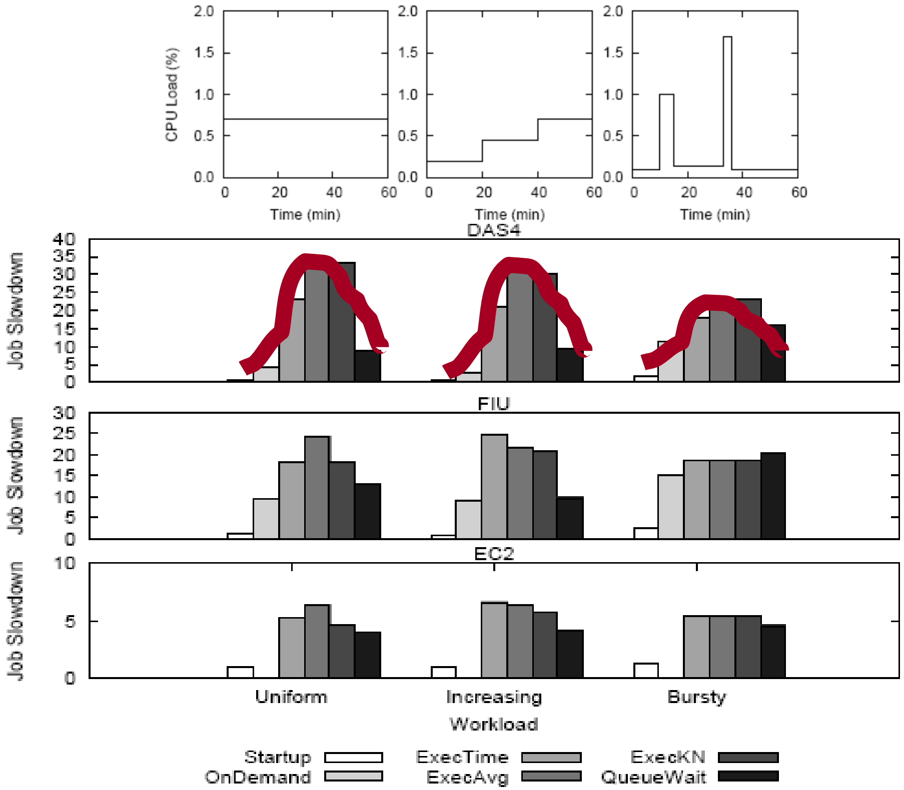
$$C_{eff}(W) = \frac{C_c(W)}{C_a(W)}$$

$$U(W) = \frac{SU_1(W)}{C_c(W)}$$

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Performance Metrics

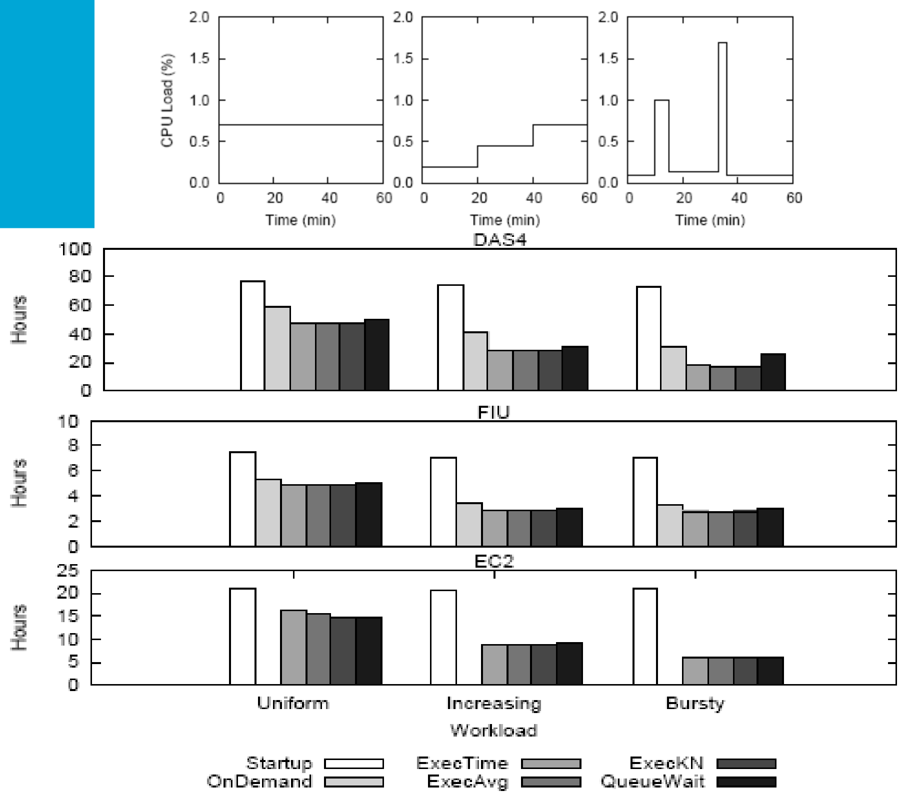




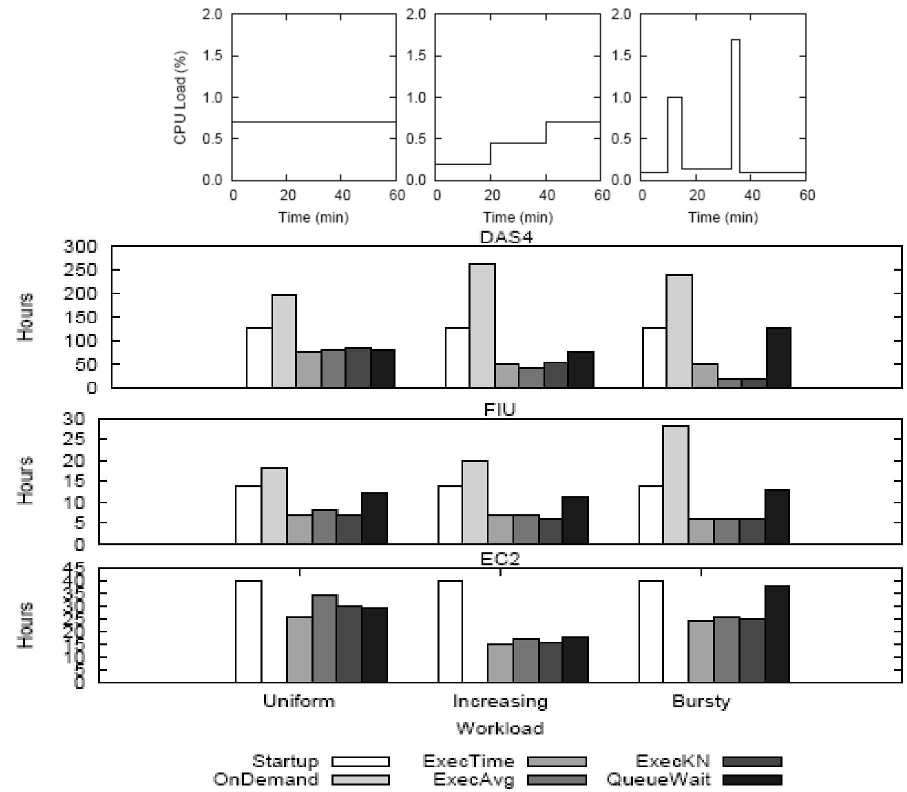
- Makespan very similar
- Very different job slowdown

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Cost Metrics



Actual Cost

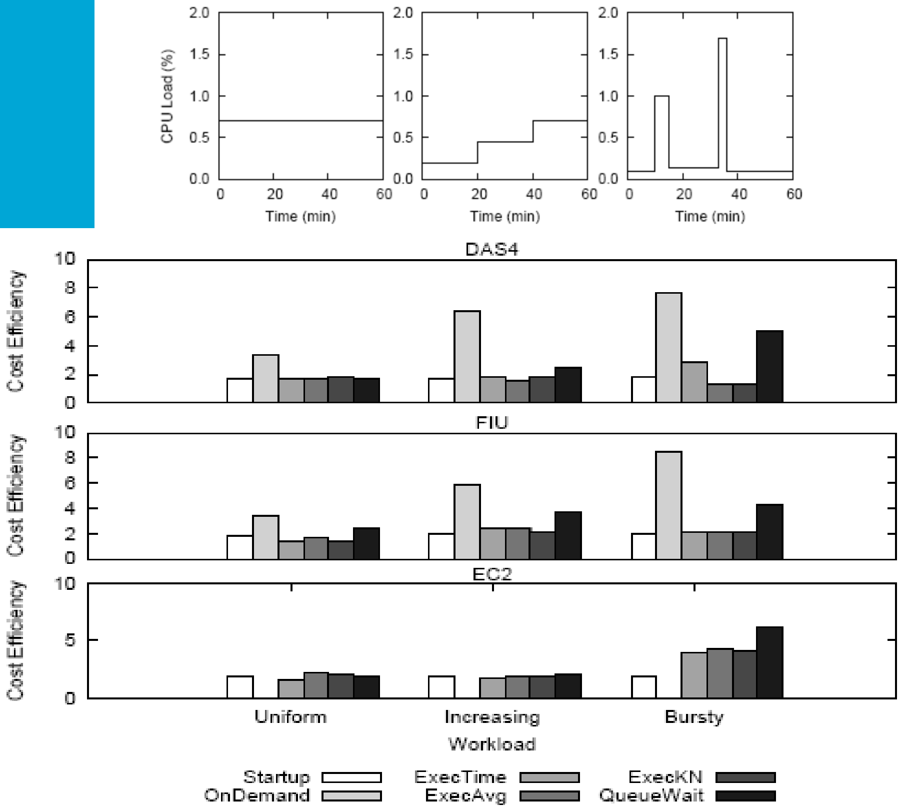


Charged Cost

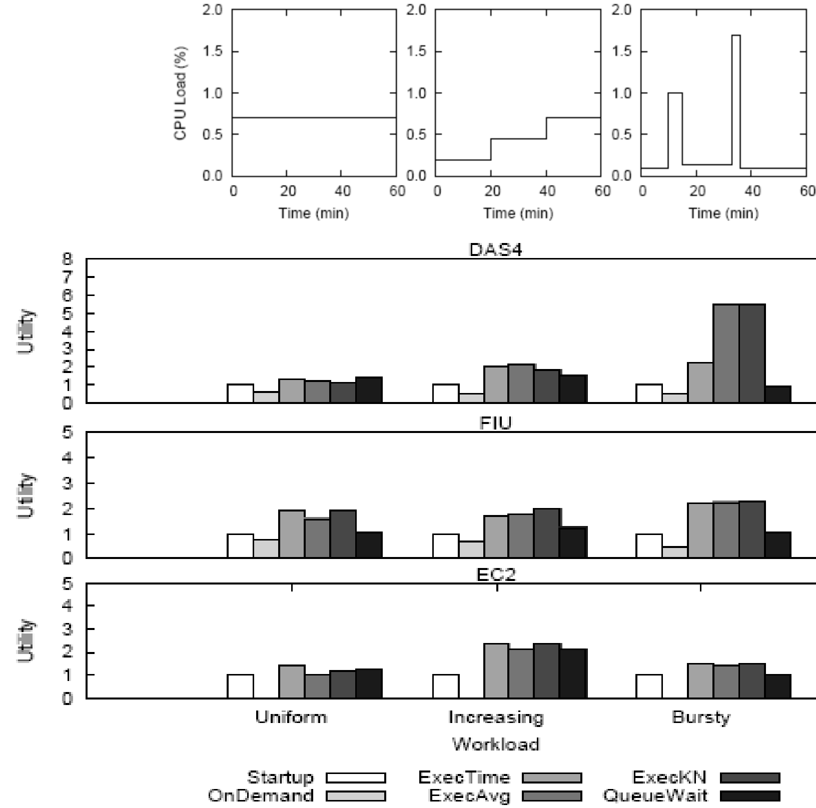
- Very different results between actual and charged
  - Cloud cost model an important selection criterion
- All policies better than Startup in actual cost
- Policies much better/worse than Startup in charged cost

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Compound Metrics



Actual Cost



Utility

- Trade-off Utility-Cost needs further investigation
- Performance or Cost, not both:
  the policies we have studied improve one, but not both

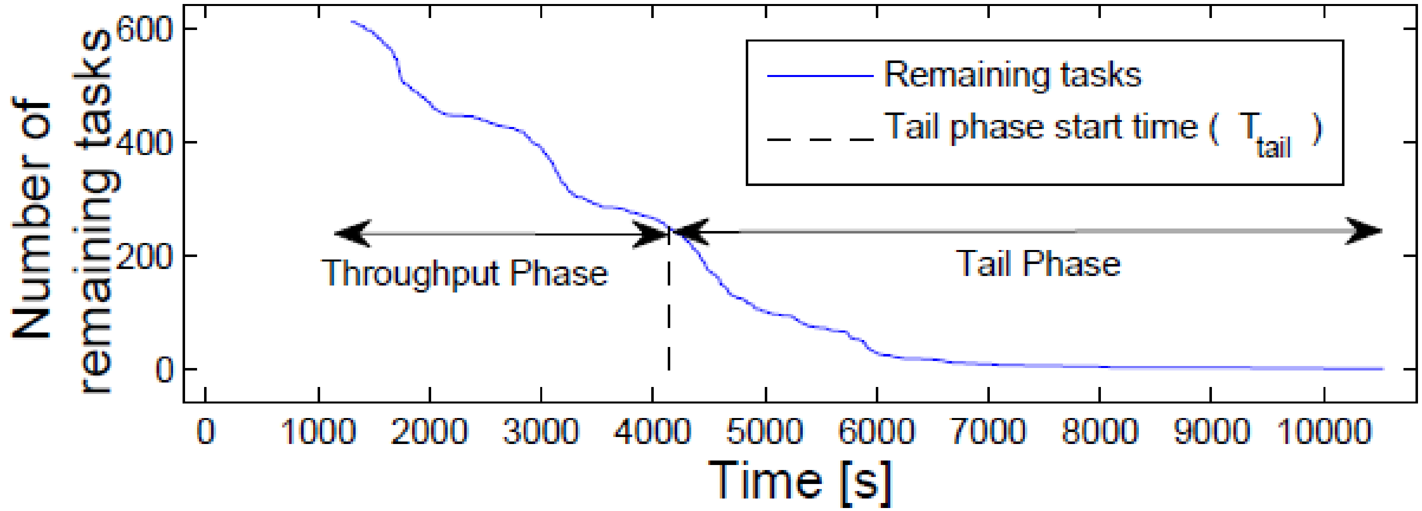
Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
   1. **Static vs IaaS**
   2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
   3. **ExPERT Pareto-Optimal User-Sched.**
   4. **Portfolio Scheduling for Data Centers**
   5. **Elastic MapReduce**
3. Take-Home Message

Helping the User Select with ExPERT: Pareto-efficient Replication of Tasks

Workload: Bags of Tasks



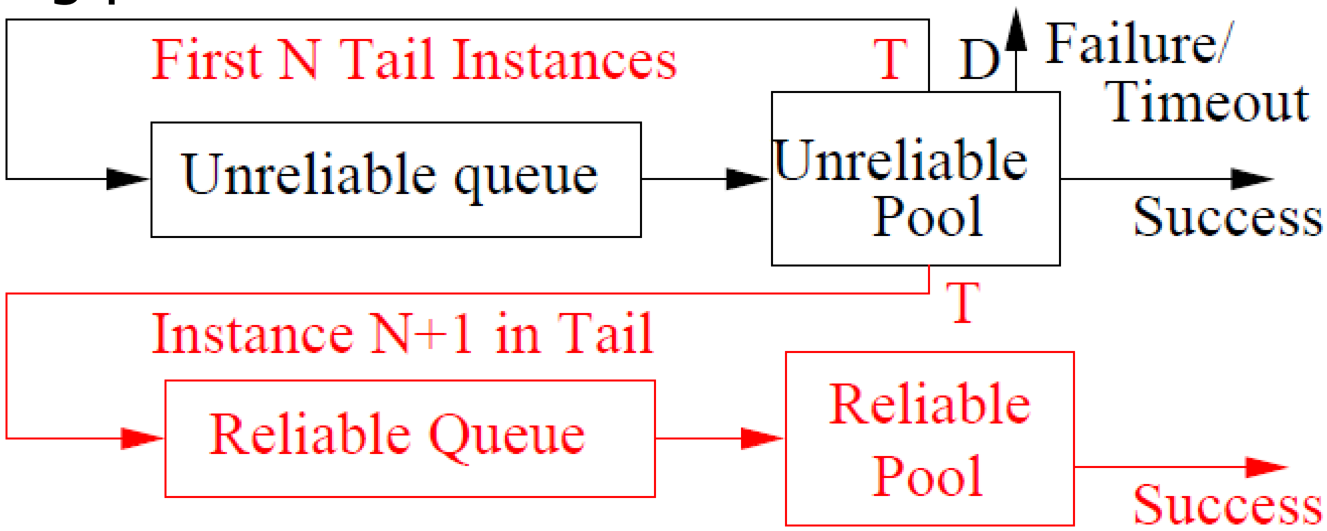
Environment

- Reliable nodes = (slow, no failure free)
- Unreliable nodes = (fast, failures, costly)

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Our Replication Mechanism

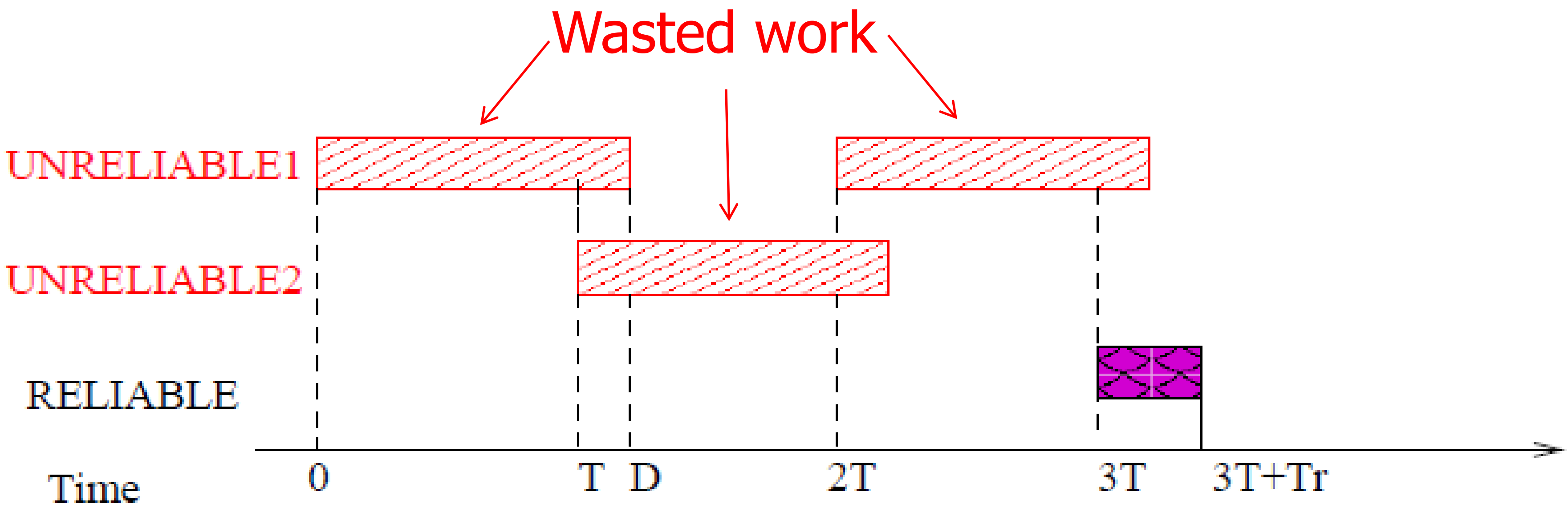
Scheduling process



Scheduling policy = (N,T,D,Mr) tuple

- D—task instance deadline
- T—when to replicate?
- N—how many times to replicate on <u>un</u>reliable?
- Nr—max ratio reliable:unreliable

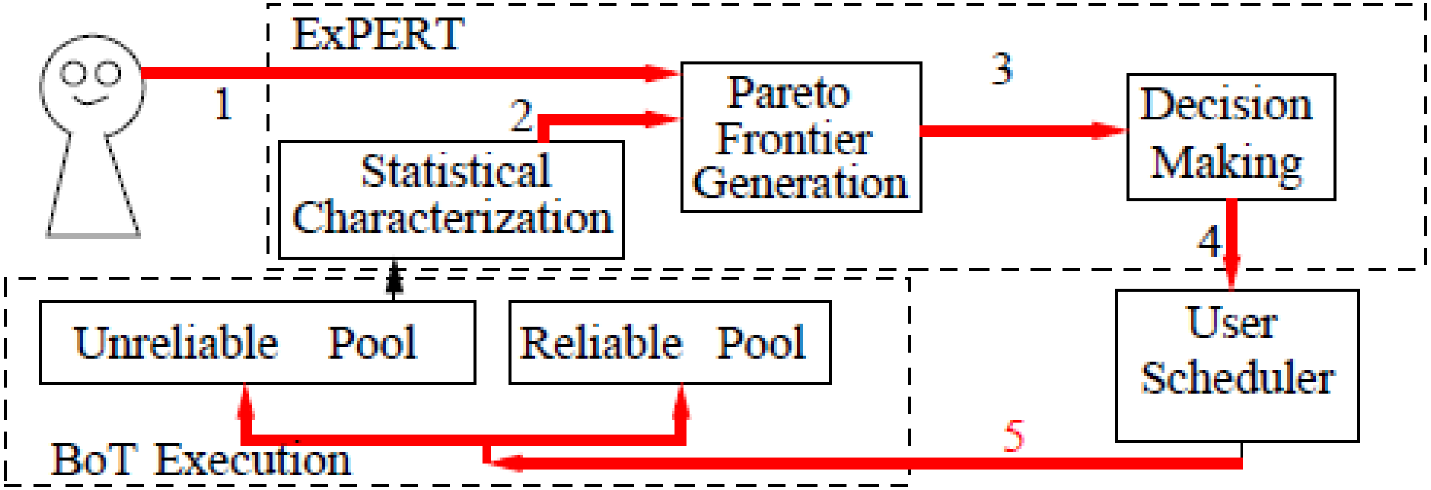
An Example with 1 Task, 2 Unreliable+1 Reliable Systems



Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

The ExPERT Policy* Recommender

* = (N,T,D,Mr) tuple

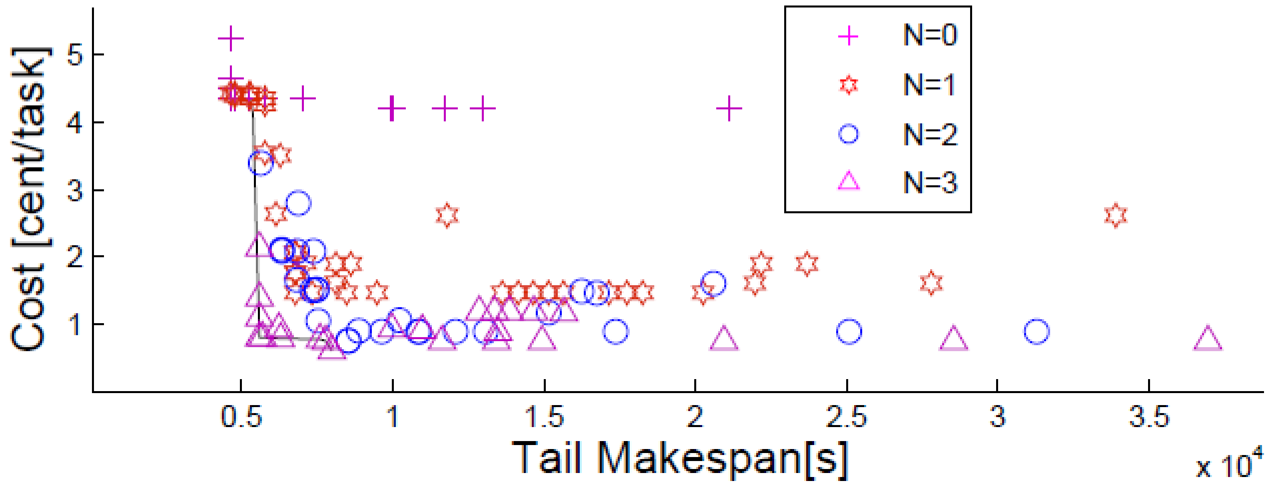


1. User specifies reliable execution time + costs
2. User provides unrealible execution statistics (failures, runtimes)
3. ExPERT computes offline a Pareto frontier of policies, <Cost>,<MS> space
   - ExPERT considers several random realizations, records average <Cost>,<MS>
4. User provides online utility functions U(<Cost>,<MS>) & ExPERT chooses online policy with best value
5. System applies policy, by applying scheduling process with selected policy

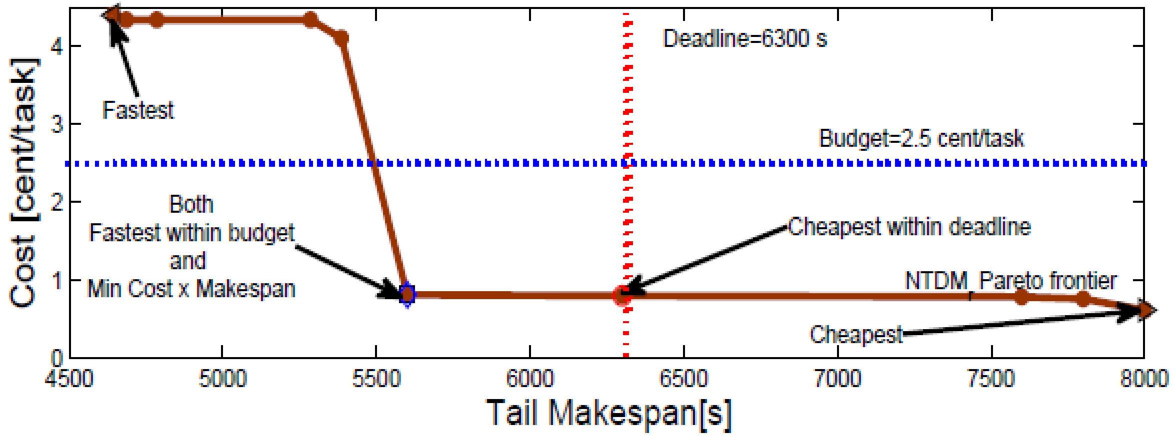
Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Anecdotal Features, Real-System Traces

- Non-Pareto (unoptimized) policies are wasteful
- Optimization non-trivial, many options



- Choice of policies at runtime: online interpretation of offline results, *point-and-click*



Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

ExPERT in Practice

Environment

Reliable Pool	Properties
Technion	20 self-owned CPUs in the Technion.
EC2	20 large EC2 cloud instances.

Unreliable Pool	Properties
UW-M	UW-Madison Condor pool (preempts).
OSG	Open Science Grid (no preemption).
UW-M + OSG	Combined: half ♯*ur* from each pool.
UW-M + EC2	Combined: 200 UW-M, 20 EC2.
UW-M + Technion	Combined: 200 UW-M, 20 Technion.

Workload

- Bioinformatics workloads, previously launched with GridBot

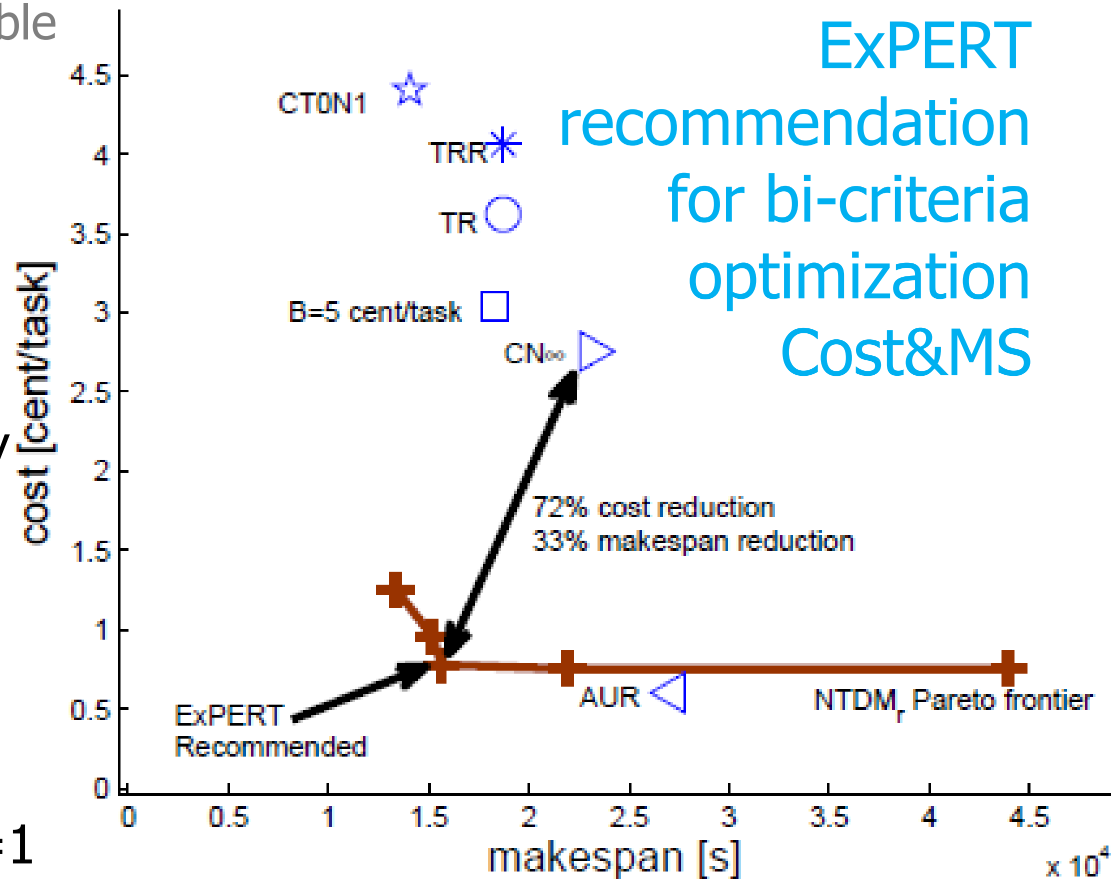
Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

ExPERT in Practice

- D—task instance deadline
- T—when to replicate?
- N—how many times to replicate on unreliable?
- Nr—max ratio reliable:unreliable

Policies

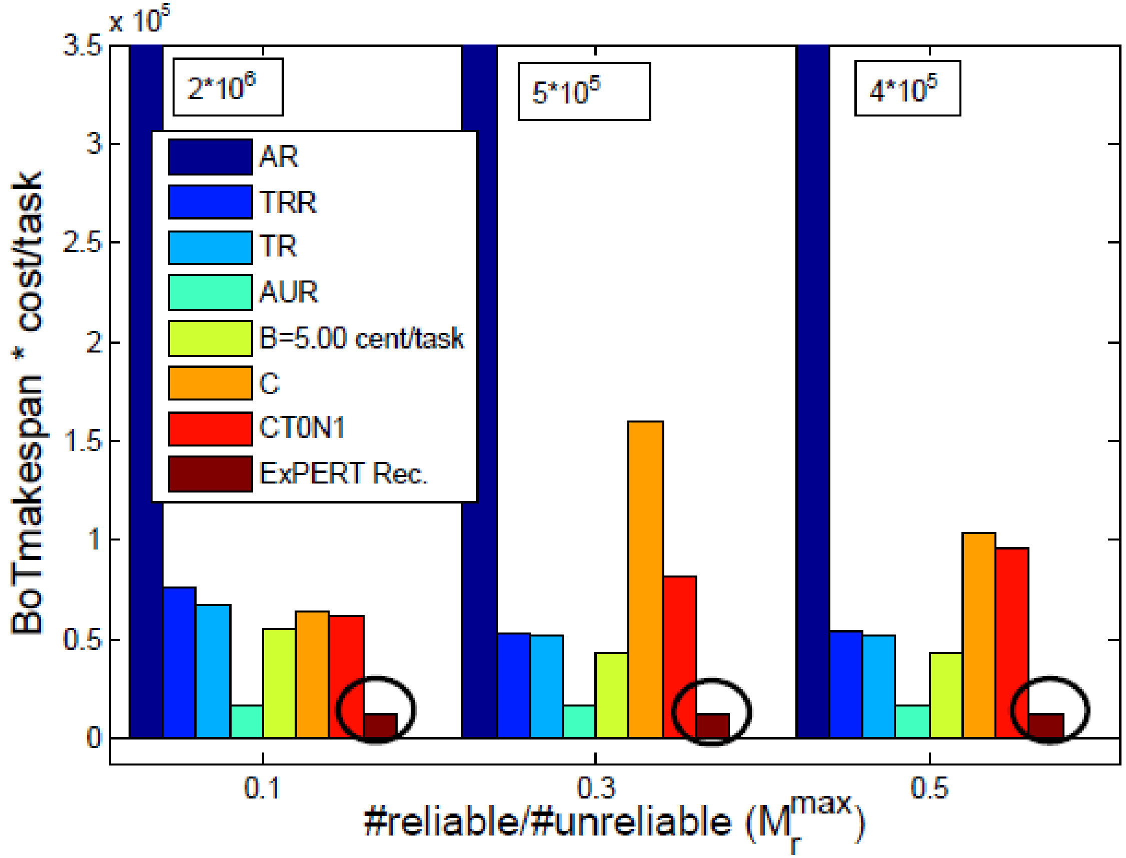
- AR—all to reliable
- AUR—all to unreliable, no replication
- TRR—Tail Replicate immediately to Reliable (N=0,T=0)
- TR—Tail to Reliable (N=0,T=D)
- CNinf—combine resources, no replication
- CT0N1—combine resources, replicate immediately at tail, N=1
- B=*cents/task—budget



ExPERT recommendation for bi-criteria optimization Cost&MS

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

ExPERT for U=Cost x MakeSpan: 25% better than 2nd-best, 72% better than 3rd-best



Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
   1. **Static vs IaaS**
   2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
   3. **ExPERT Pareto-Optimal User-Sched.**
   4. **Portfolio Scheduling for Data Centers**
   5. **Elastic MapReduce**
3. Take-Home Message

Warm-Up Question:
(2 minutes think-time + 2 minutes open discussion)

- Think about own experience
- Convince your partner before proposing an answer
- Tell everyone the answer

Q: What are the major issues of scheduling various types of workloads in current data centers?

What I'll Talk About

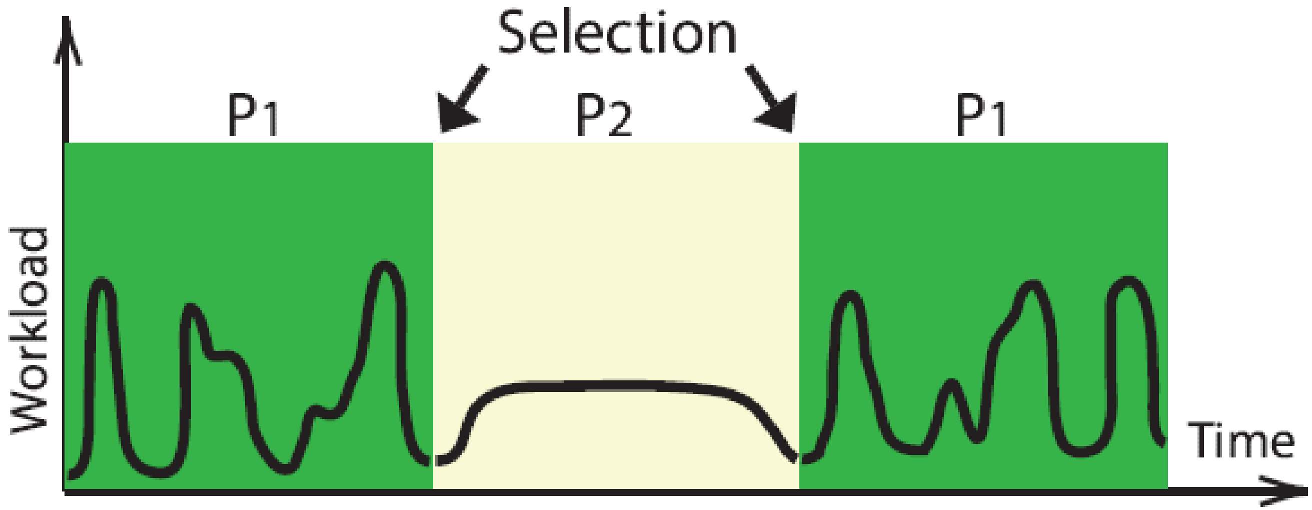
1. Why portfolio scheduling?
2. What is portfolio scheduling? In a nutshell…
3. **Our periodic portfolio scheduler for the data center**
    1. **Operational model**
    2. **A portfolio scheduler architecture**
    3. **The creation and selection components**
    4. **Other design decisions**
4. Experimental results
   How useful is our portfolio scheduler? How does it work in practice?
5. Our ongoing work on portfolio scheduling
6. How novel is our portfolio scheduler? A discussion about related work
7. Conclusion

Why Portfolio Scheduling?

- **Data centers increasingly popular**
  - Constant deployment since mid-1990s
  - Users moving their computation to IaaS clouds
  - Consolidation efforts in mid- and large-scale companies
- **Old scheduling aspects**
  - Hundreds of approaches, each targeting specific conditions—which?
  - No one-size-fits-all policy
- **New scheduling aspects**
  - New workloads
  - New data center architectures
  - New cost models

- **Developing a scheduling policy is risky and ephemeral**
- **Selecting a scheduling policy for your data center is difficult**

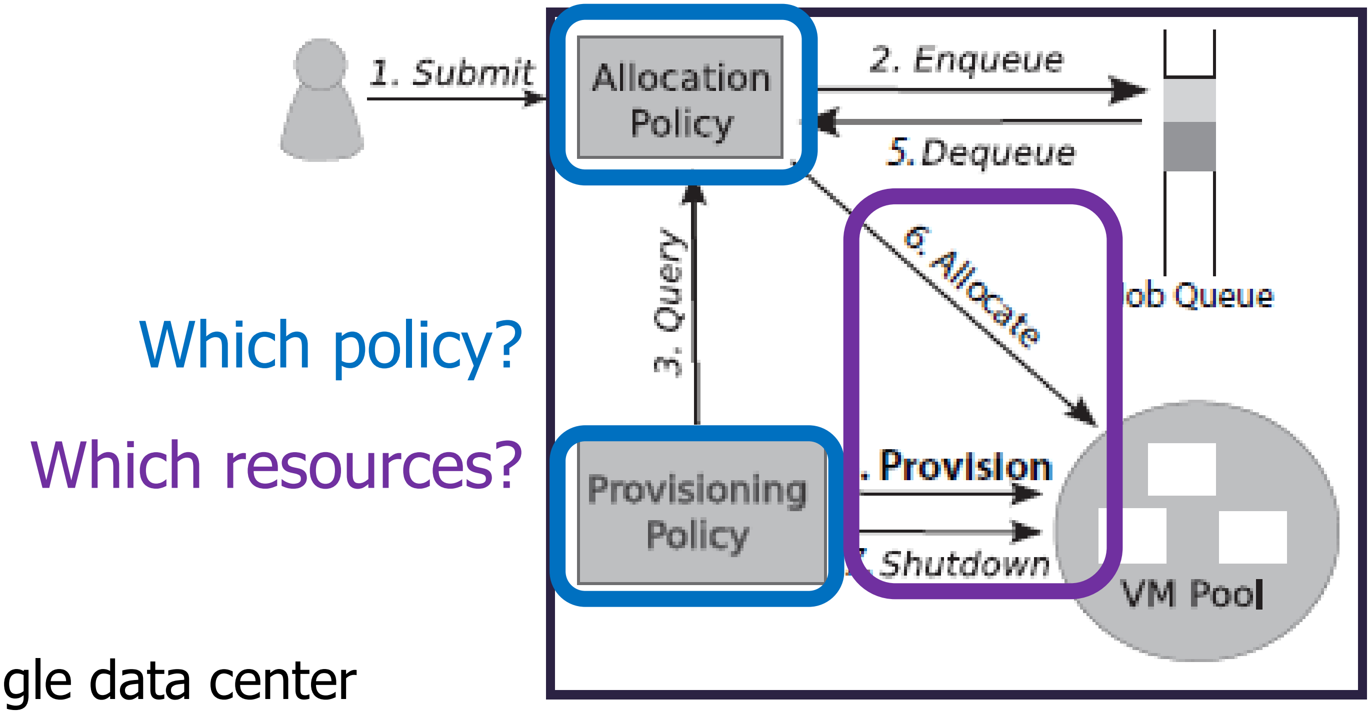
What is Portfolio Scheduling? In a Nutshell, for Data Centers



- Create a set of scheduling policies
  - Resource provisioning and allocation policies, in this work
- Online selection of the active policy, at important moments
  - Periodic selection, in this work
- Same principle for other changes: pricing model, system, …

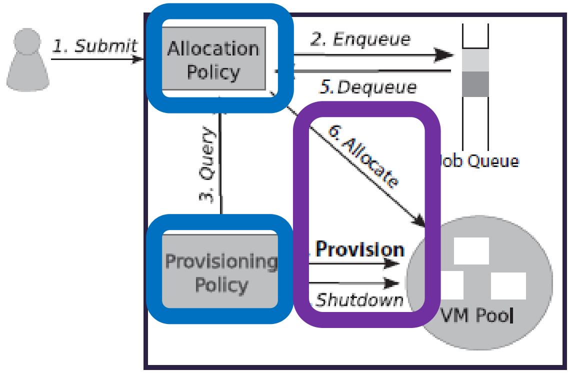
Background Information

Operational Model



- Single data center
- VM pool per user
- Provisioning and allocation of resources via policies
- Issues orthogonal to this model: failures, pre-emption, migration, …

Portfolio Scheduling
The Process



Which policies to include?

Which policy to activate?

Creation

Selection

Application

Reflection

Which changes to the portfolio?

Which resources? What to log?

Portfolio Scheduling Components
Creation

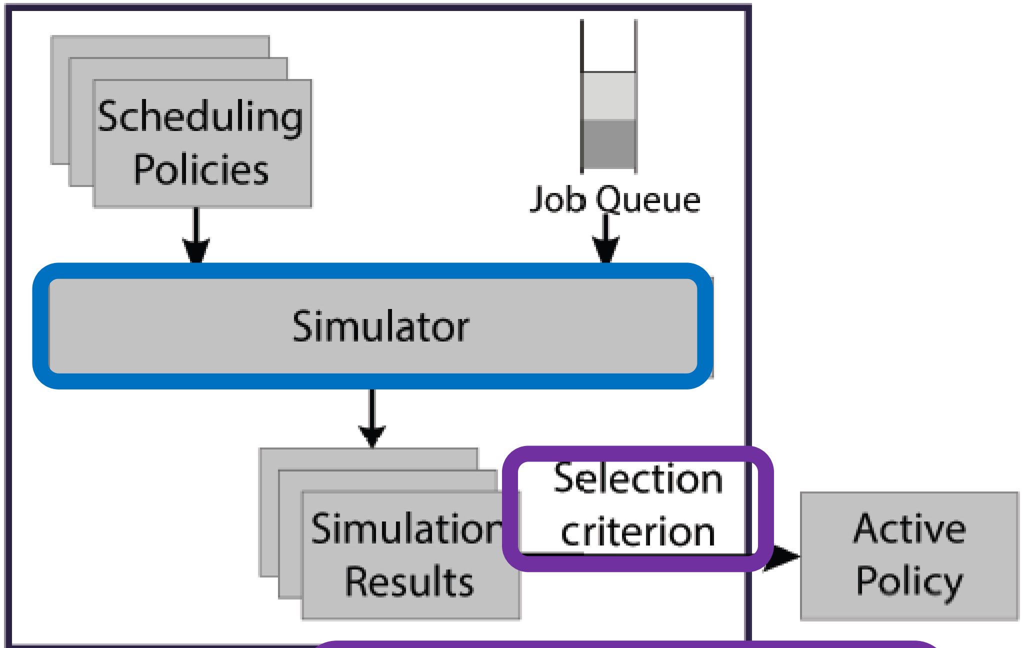
- Scheduling policy = (provisioning, job selection) tuple
  - We assume in this work all VMs are equal and exclusively used (no VM selection policy—we study these in other work)

- Provisioning policies
  - Start-Up: all resources available from start to finish of execution (classic)
  - On-Demand, Single VM (ODS): one new VM for each queued job
  - On-Demand, Geometric (ODG): grow-shrink exponentially
  - On-Demand, Execution Time (ODE): lease according to estimation of queued runtime (uses historical information and a predictor) $\alpha^0, \alpha^1, \ldots, \alpha^n$
  - On-Demand, Wait Time (ODW): leases only for jobs with high wait times
  - On-Demand, XFactor (ODX): tries to ensure constant slowdown, via observed wait time and estimated run time
- Job selection policies
  - FCFS, SJF (assumes known or well-estimated run-times)

Deng, Song, Ren, and Iosup. Exploring Portfolio Scheduling for Long-term Execution of Scientific Workloads in IaaS Clouds. Submitted to SC|13.

Portfolio Scheduling Components
Selection

- Periodic execution
- Simulation-based selection
- Utility function
- Alternatives simulator
  - Expert human knowledge
  - Running workload sample in similar environment, under different policies
  - mathematical analysis
- Alternatives utility function
  - Well-known and exotic functions



$$U = \kappa \cdot \left(\frac{R_J}{R_V}\right)^{\alpha} \cdot \left(\frac{1}{S}\right)^{\beta}$$

R_J: Total Runtime of Jobs

R_V: Total Runtime of VMs

S: Slowdown

$\alpha=\beta=1$

K=100

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Setup
Simulator and Metrics

- The DGSim simulator
  - Since 2007
  - Scheduling in single- and multi-cluster grids
  - Scheduling in IaaS clouds

Iosup, Sonmez, Epema. DGSim: Comparing Grid Resource Management Architectures through Trace-Based Simulation. Euro-Par 2008.

- Metrics
  - Average Job Wait-Time
  - Average Job Slowdown
  - Resource utilization
  - Charged Cost
  - Utility

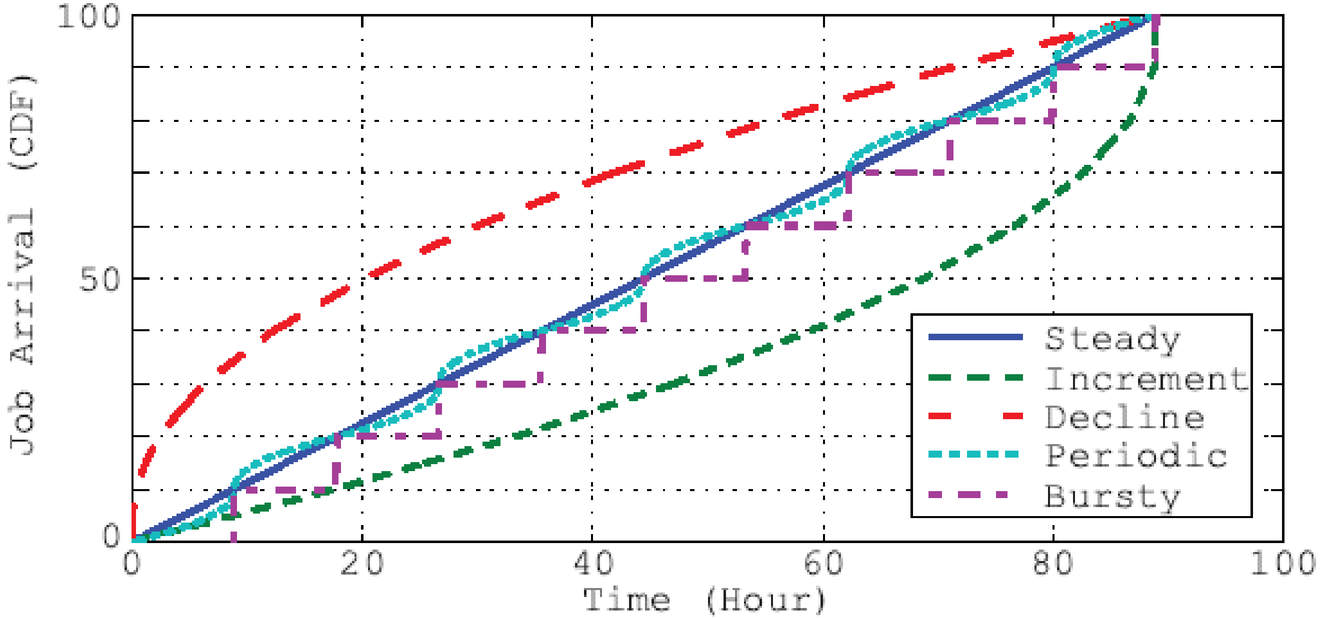
$$C_c(W) = \sum_{i \,\in\, leased\ VMs} \lceil t_{stop}(i) - t_{start}(i) \rceil$$

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Setup

Synthetic and Real Traces

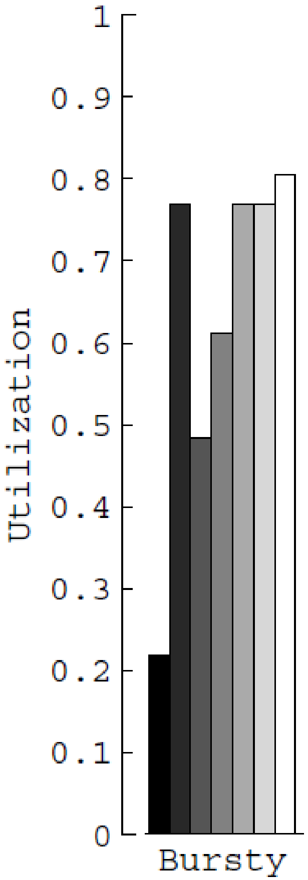
- Synthetic Workloads: 5 arrival patterns

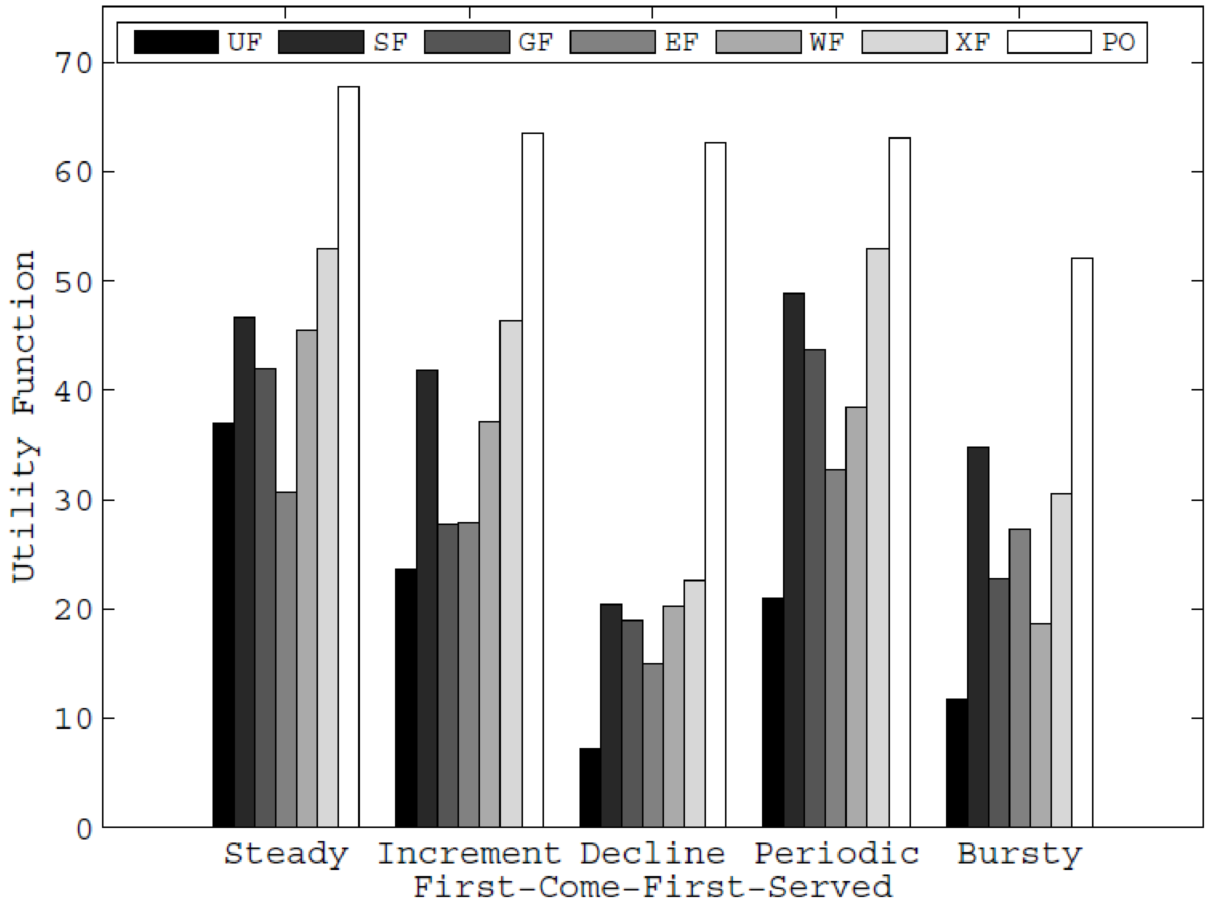


- Real Trace: ANL Intrepid 2009
  - 8 months
  - 68,936 jobs

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Results, **Synthetic** Workloads Resource Utilization + Workload Utility



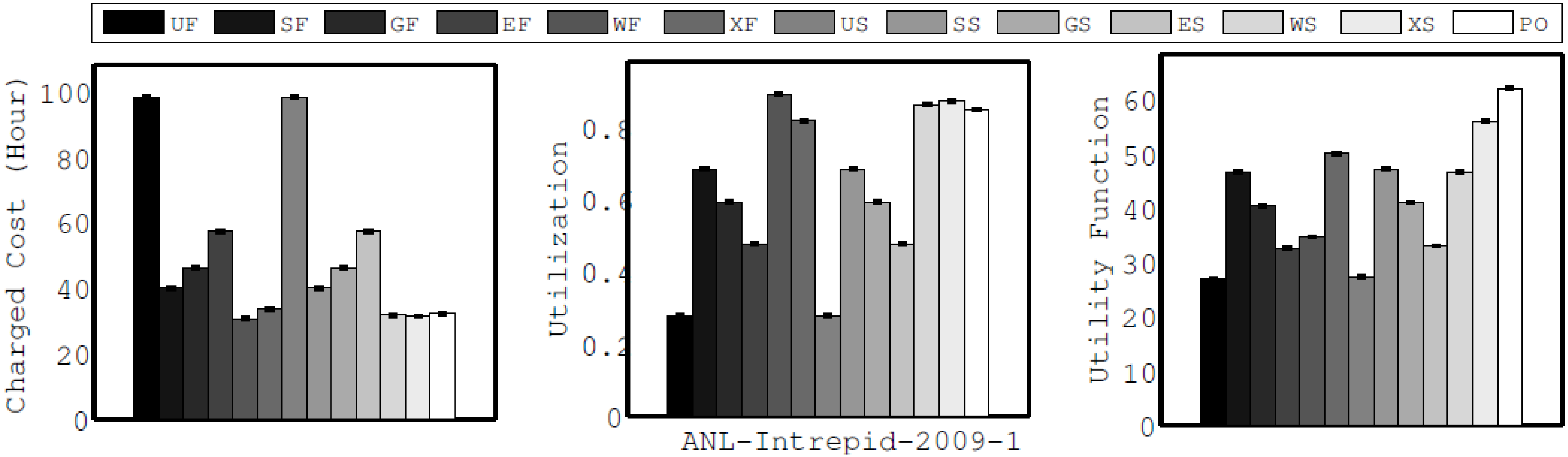


- POrtfolio leads to high utilization
- Start-Up leads to poor utilization
- POrtfolio leads to better utility
- Start-Up leads to poor utility

TU Delft

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Results, **ANL Intrepid** Workload
Cost + Utilization + Utility

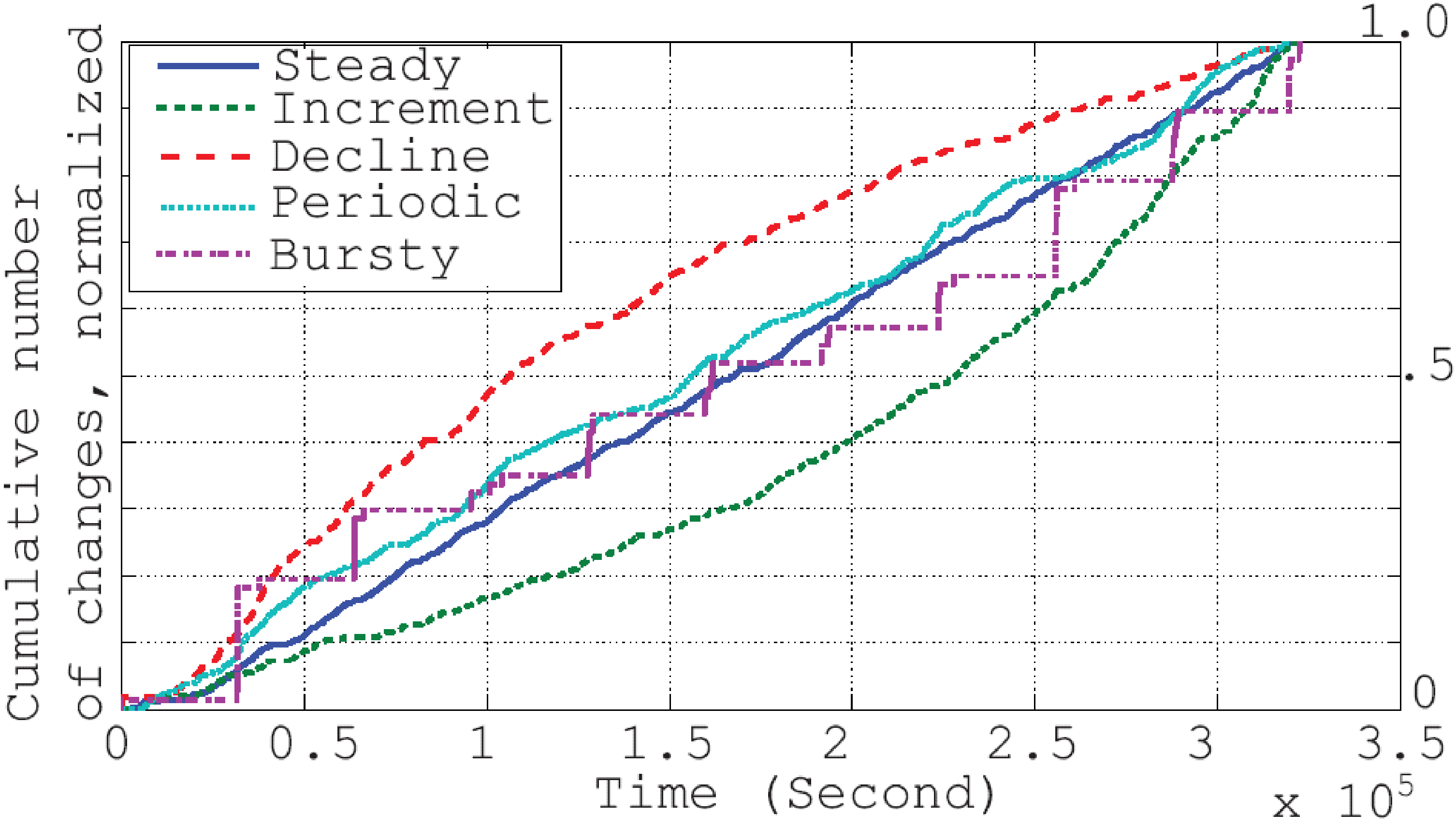


- POrtfolio not best for each metric
- POrtfolio leads to low cost
- POrtfolio leads to high utilization
- POrtfolio leads to high utility (slowdown-utilization compound)

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Results

Operation of the Portfolio Scheduler

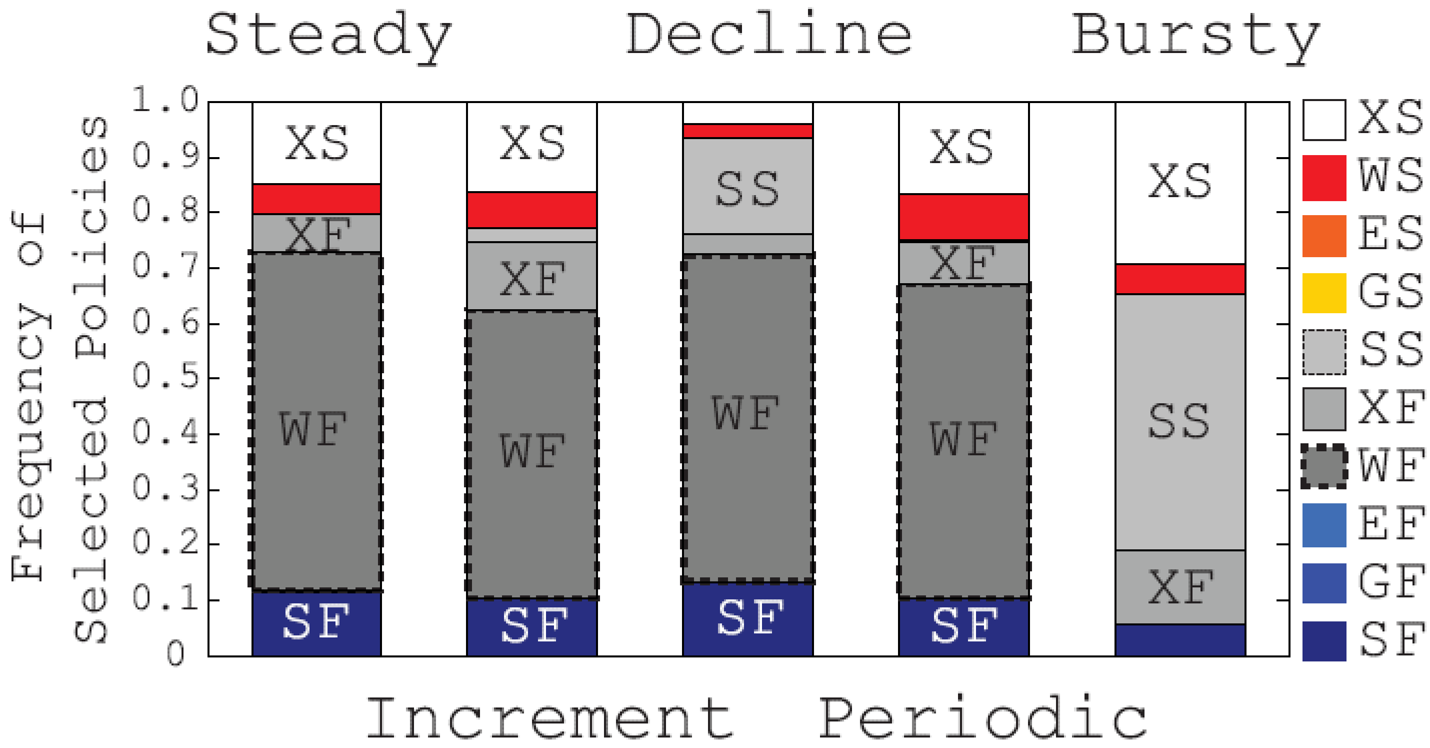


- Policy change follows arrival pattern
- ANL-Intrepid between Steady and Periodic

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Experimental Results

Operation of the Portfolio Scheduler



- No single policy is always selected for the same workload
- Different workloads, different top-3 policies

Deng, Verboon, Iosup. A Periodic Portfolio Scheduler for Scientific Computing in the Data Center. JSSPP'13.

Portfolio Scheduling for Online Gaming
(also for Scientific Workloads)

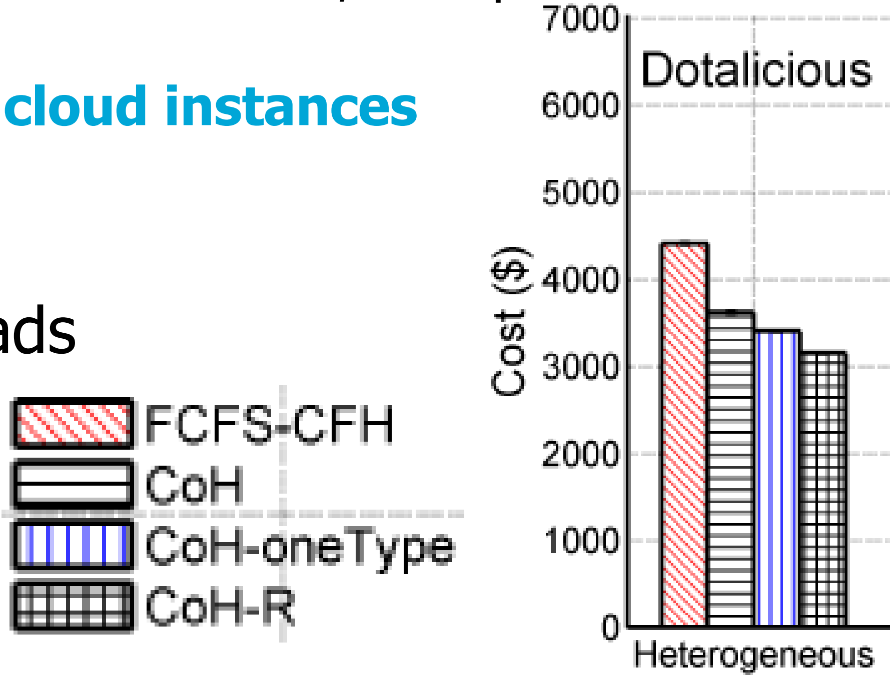
CoH = Cloud-based, online, Hybrid scheduling

- Intuition: keep rental cost low by finding good mix of machine configurations and billing options
- Main idea: **portfolio scheduler** = run *both* solver of an Integer Programming Problem and various heuristics, then pick best schedule at deadline
- Additional feature: Can use **reserved cloud instances**

- Promising early results, for

  Gaming (and scientific) workloads

Trace	#jobs	average runtime [s]
Grid5000	200,450	2728
LCG	188,041	8971
DotaLicious	109,251	2231



Shen, Deng, Iosup, and Epema. Scheduling Jobs in the Cloud Using On-demand and Reserved Instances, EuroPar'13.

Related Work

- Computational portfolio design
  - Huberman'97, Streeter et al.'07 '12, Bougeret'09, Goldman'12, Gagliolo et al.'06 '11, **Ohad Shai et al. JSSPP'13 (please attend!)**
  - We focus on dynamic, scientific workloads
  - We use an utility function that combines slowdown and utilization

- Modern portfolio theory in finance
  - Markowitz'52, Magill and Constantinides'76, Black and Scholes'76
  - Dynamic problem set vs fixed problem set
  - Different workloads and utility functions
  - Selection and Application very different

- Historical simulation
- General scheduling

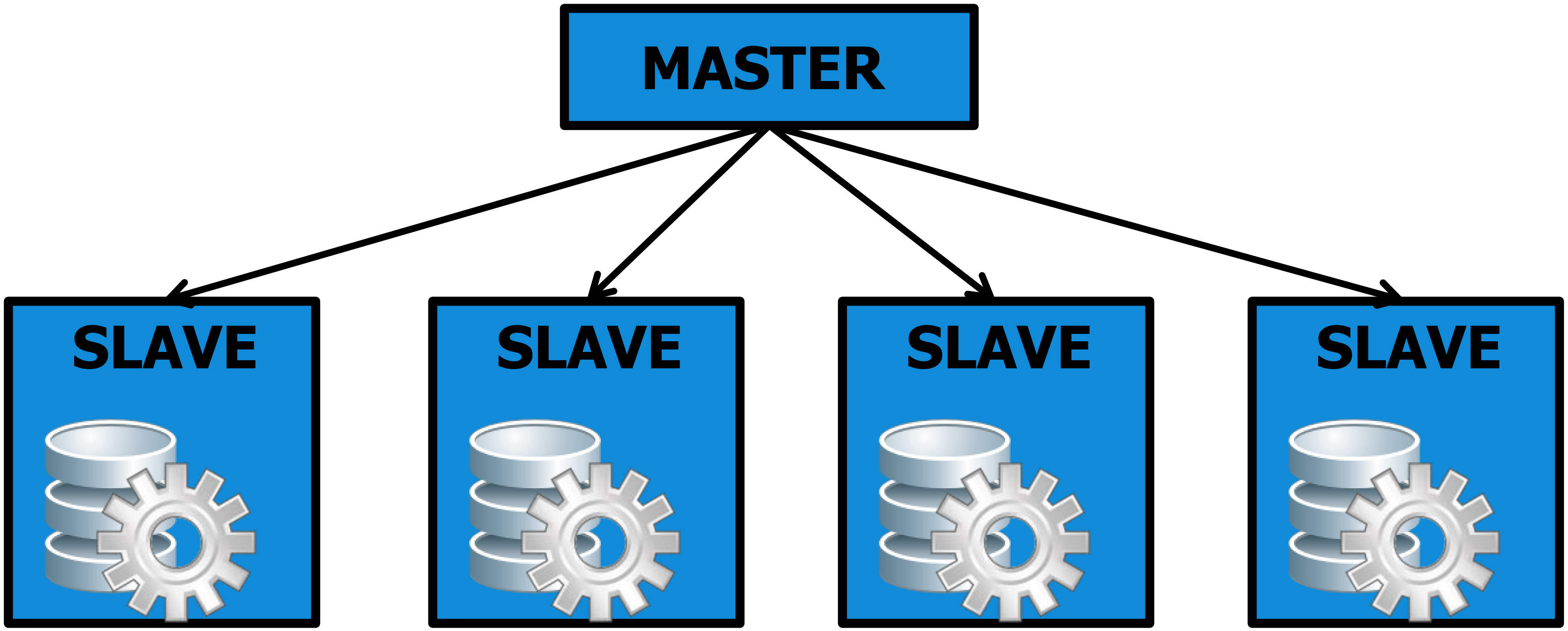
- Hyper-scheduling, meta-scheduling
  - The learning rule may defeat the purpose, via historical bias to dominant policy
  - Dynamic selection and reflection processes

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
   1. **Static vs IaaS**
   2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
   3. **ExPERT Pareto-Optimal User-Sched.**
   4. **Portfolio Scheduling for Data Centers**
   5. **Elastic MapReduce**
3. Take-Home Message

MapReduce Overview

- MR cluster
  - Large-scale data processing
  - Master-slave paradigm
- Components
  - Distributed file system (storage)
  - MapReduce framework (processing)



Warm-Up Question: (2 minutes think-time + 2 minutes open discussion)

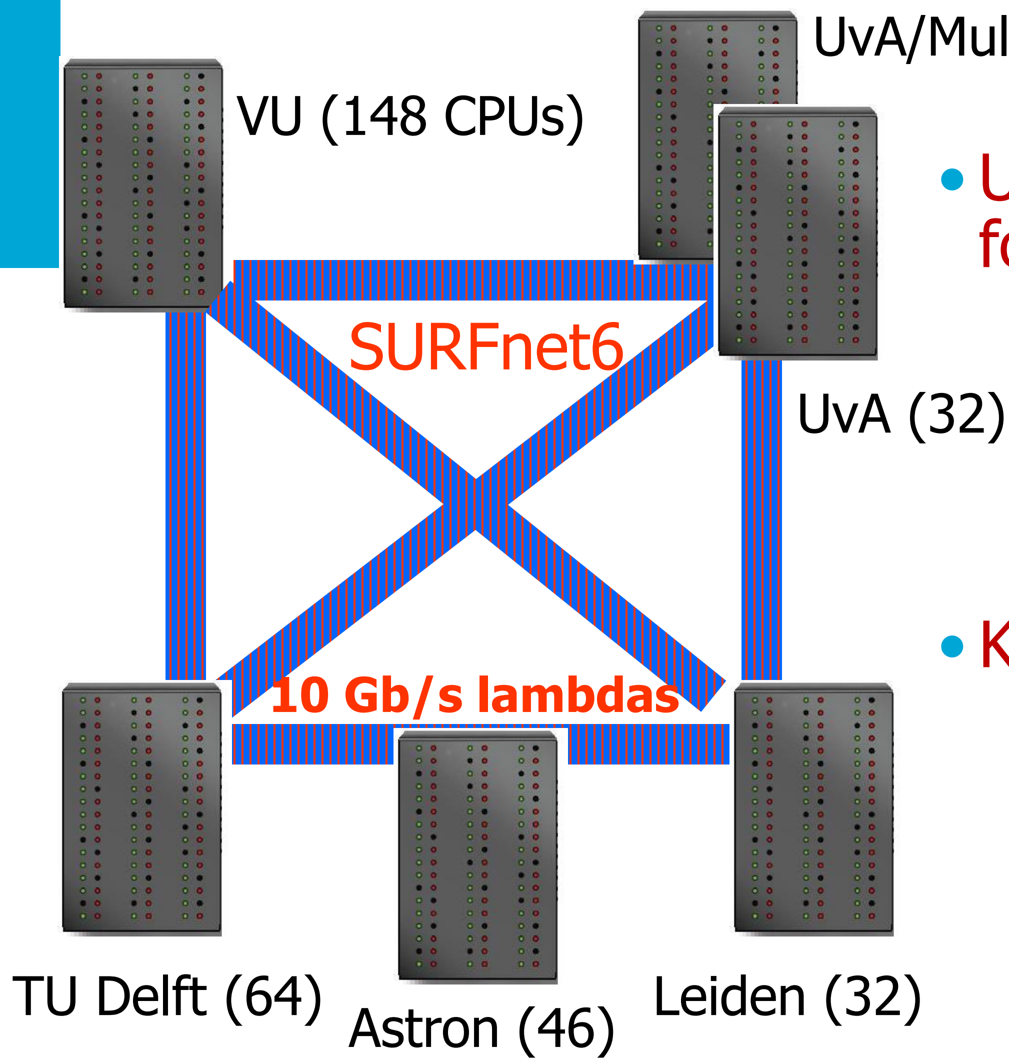
- Think about own experience
- Convince your partner before proposing an answer
- Tell everyone the answer

Q: How would **you** make use of IaaS clouds to run MapReduce workloads? (What new mechanisms, algorithms, systems are required?)

What I'll Talk About?

1. MapReduce in the DAS
2. **Our Elastic MapReduce**
    1. **Main idea: the growth-shrink mechanism**
    2. **Several policies**
3. Experimental setup
4. Experimental results

The DAS-4 Infrastructure



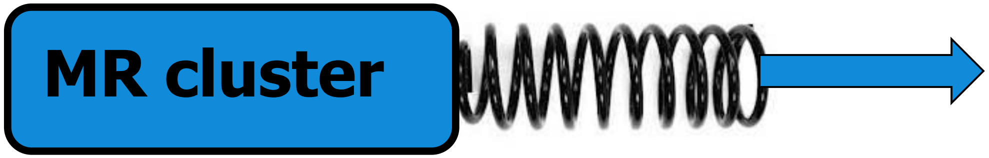
- Used for research in systems for over a decade
  - 1,600 cores (quad cores)
  - 2.4 GHz CPUs, GPUs
  - 180 TB storage
  - 10 Gbps Infiniband
  - 1 Gbps Ethernet
- Koala grid scheduler

Why Dynamic MapReduce Clusters?

- Improve resource utilization
  - Grow when the workload is too heavy
  - Shrink when resources are idle
- Fairness across multiple MR clusters
  - Redistribute idle resources
  - Allocate resources for new MR clusters

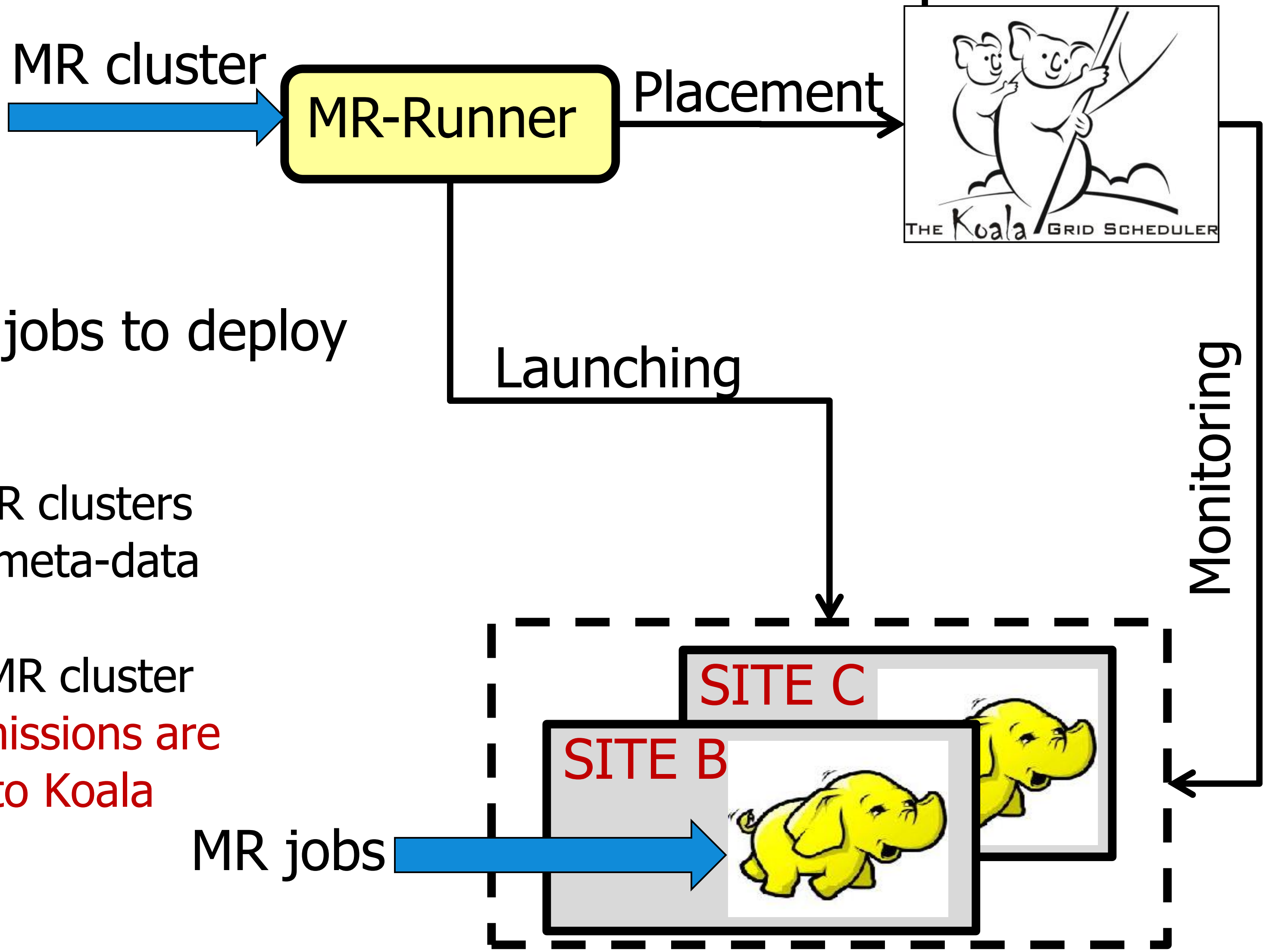
Isolation

- Performance
- Failure
- Data
- Version



Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

KOALA Grid Scheduler and MapReduce



- Users submit jobs to deploy MR clusters
- Koala
  - Schedules MR clusters
  - Stores their meta-data
- MR-Runner
  - Installs the MR cluster
  - MR job submissions are transparent to Koala

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

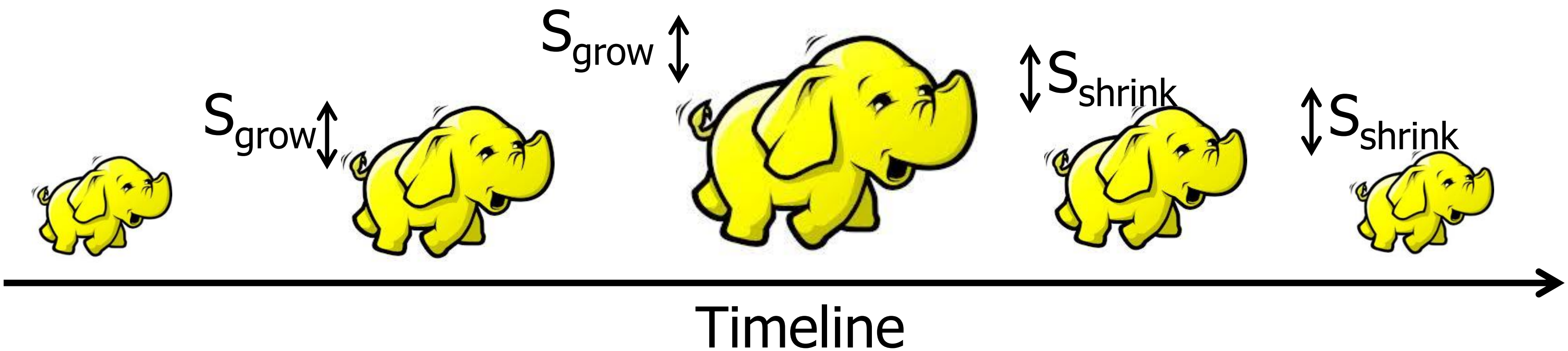
System Model

- Two types of nodes
  - Core nodes: TaskTracker and DataNode
  - Transient nodes: only TaskTracker

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Resizing Mechanism

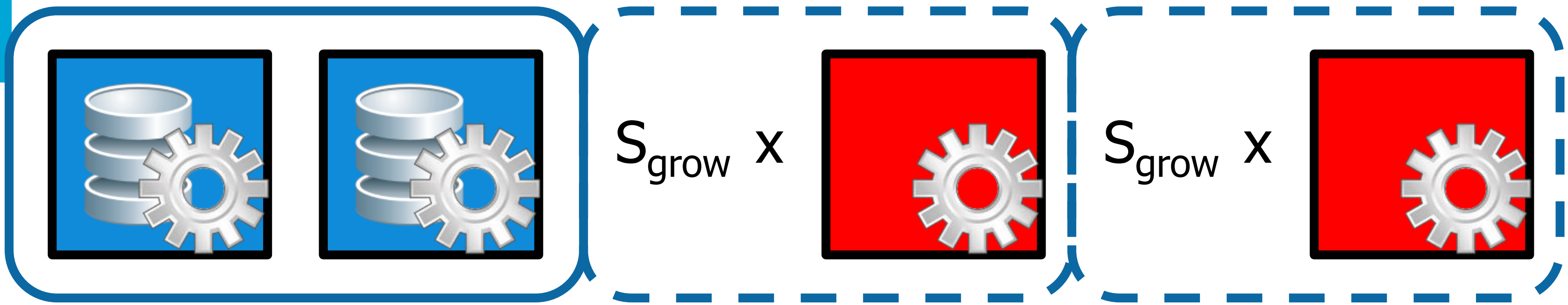
- Two-level provisioning
  - Koala makes resource offers / reclaims
  - MR-Runners accept / reject request
- Grow-Shrink Policy (GSP)
  - MR cluster utilization:
  - $$F_{\min} \le \frac{totalTasks}{availSlots} \le F_{\max}$$
  - Size of grow and shrink steps: $\mathbf{S_{grow}}$ and $\mathbf{S_{shrink}}$



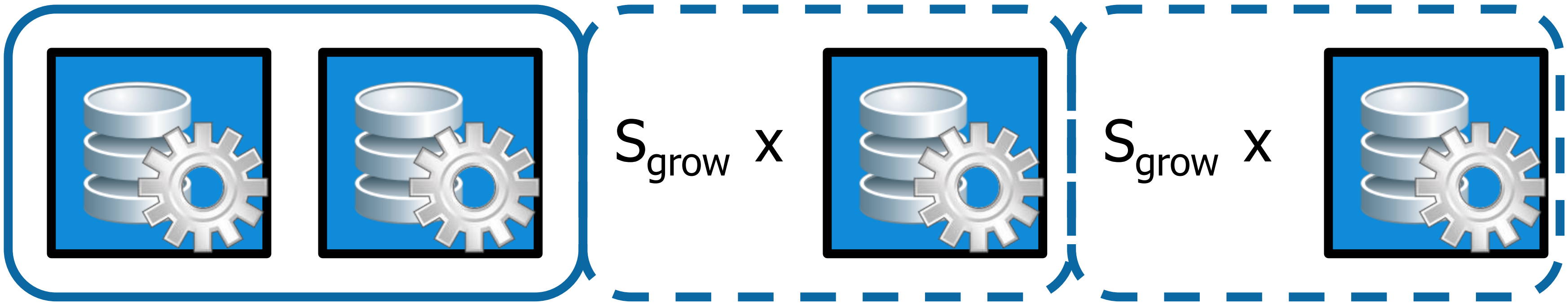
Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Baseline Policies

- Greedy-Grow Policy (GGP)—only grow with transient nodes:



- Greedy-Grow-with-Data Policy (GGDP)—grow, core nodes:



Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Setup

- *98% of jobs @ Facebook take less than a minute*
- *Google reported computations with TB of data*

- DAS-4
- Two applications: Wordcount and Sort

Workload 1

- Single job
- 100 GB
- Makespan

Workload 2

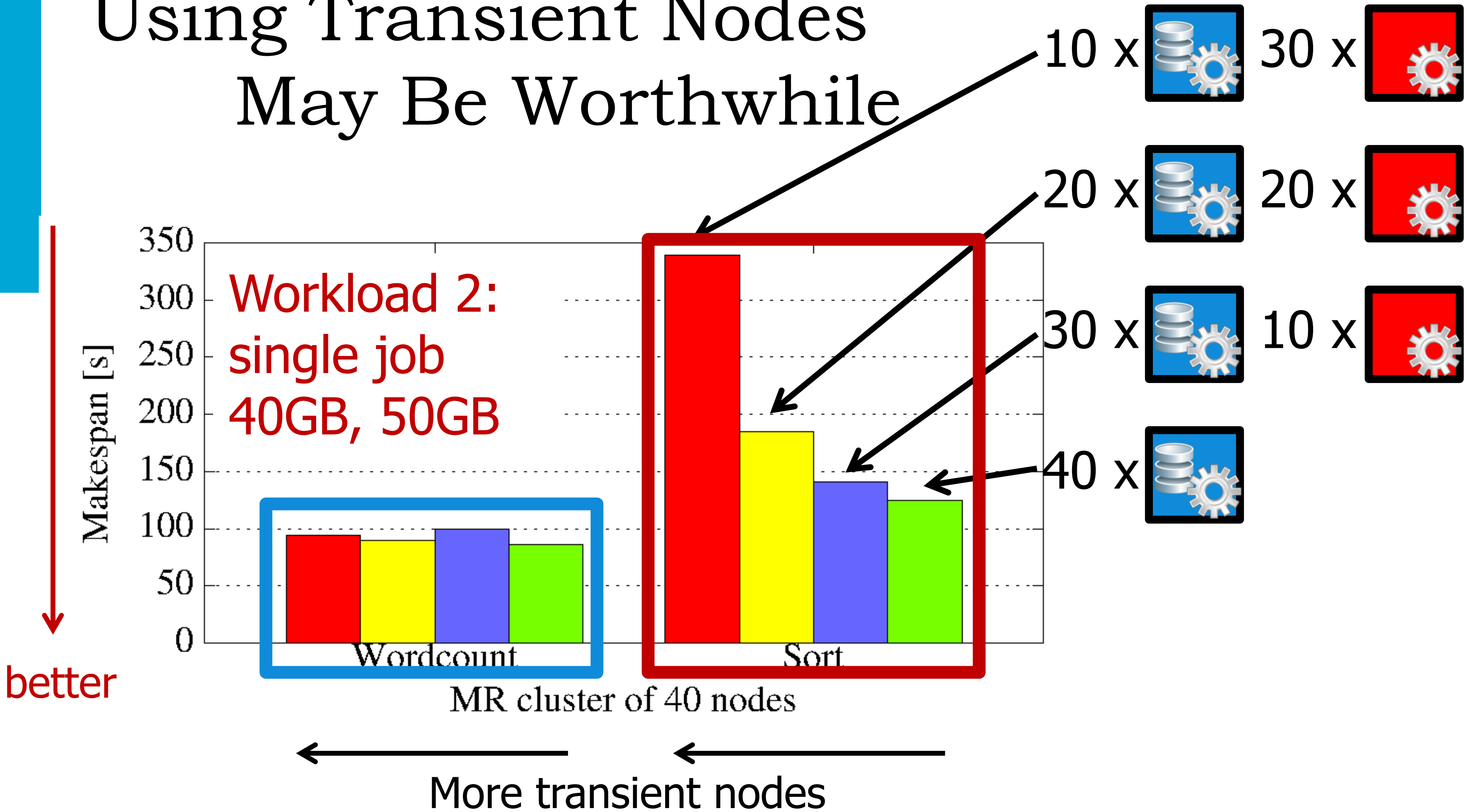
- Single job
- 40 GB, 50 GB
- Makespan

Workload 3

- Stream of 50 jobs
- 1 GB → 50 GB
- Average job execution time

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Using Transient Nodes May Be Worthwhile

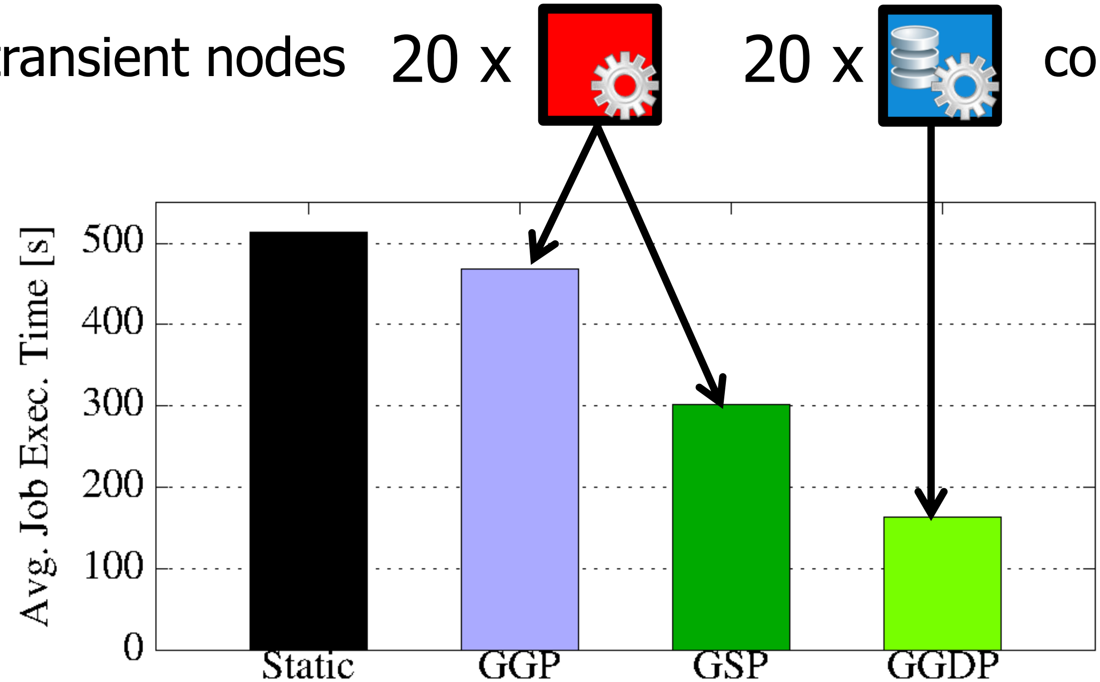


- Replacing more core with transient nodes works for Wordcount
- Wordcount scales better than Sort on transient nodes

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Resizing using Core or Transient Nodes vs Static Worthwhile

transient nodes 20 x … 20 x … core nodes



better

Workload 3 => 20 x 
50 jobs
1—50 GB

- Resizing bounds
  $F_{min} = 0.25$
  $F_{max} = 1.25$
- Resizing steps
  - GSP
    $S_{grow} = 5$
    $S_{shrink} = 2$
  - GG(D)P
    $S_{grow} = 2$

Ghit and Epema. Resource Management for Dynamic MapReduce Clusters in Multicluster Systems. MTAGS 2012. Best Paper Award.

Agenda

1. Introduction to IaaS Cloud Scheduling
2. **PDS Group Work on Cloud Scheduling**
   1. **Static vs IaaS**
   2. **IaaS Cloud Scheduling, an empirical comparison of heuristics**
   3. **ExPERT Pareto-Optimal User-Sched.**
   4. **Portfolio Scheduling for Data Centers**
   5. **Elastic MapReduce**
3. Take-Home Message

~~Conclusion~~ Take-Home Message

- http://www.st.ewi.tudelft.nl/~iosup/
- http://www.pds.ewi.tudelft.nl/

- A.Iosup@tudelft.nl
- DengKefeng@nudt.edu

- **Comparison static vs IaaS cloud environements**
- **Performance of provisioning and allocation policies for Ia ls**
  - **No single policy works best in all settings**
- **Automatic**
  - **ExPERT: Pareto-optimal selection on users' behalf**

Alexandru Iosup

- **Portfolio Scheduling = set of scheduling policies, online selection**
  - **Creation, Selection, Application,** Reflection
  - **Periodic portfolio scheduler for data centers**
- **Elastic MapReduce (PDS team)**

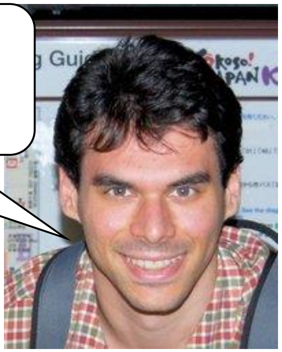
Thank you for your attention! Questions? Suggestions? Observations?

More Info:

HPDC 2013

- http://www.st.ewi.tudelft.nl/~iosup/research.html
- http://www.st.ewi.tudelft.nl/~iosup/research_cloud.html
- http://www.pds.ewi.tudelft.nl/

Do not hesitate to contact me…



Alexandru Iosup

A.Iosup@tudelft.nl
http://www.pds.ewi.tudelft.nl/~iosup/ (or google "iosup")
Parallel and Distributed Systems Group
Delft University of Technology